175 PTAS

71



CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Editorial Delta, S.A.

MI COMPUTER CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona

Volumen VI-Fascículo 71

Director:	José Mas Godayol
Director editorial:	Gerardo Romero
Jefe de redacción:	Pablo Parra
Coordinación editorial:	Jaime Mardones
	Francisco Martín
Asesor técnico:	Ramón Cervelló

Redactores y colaboradores: G. Jefferson, R. Ford, F. Martín, S. Tarditti, A. Cuevas, F. Blasco

Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Paseo de Gracia, 88, 5.°, 08008 Barcelona
Tels. (93) 215 10 32 / (93) 215 10 50 - Télex 97848 EDLTE

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador*, se publica en forma de 96 fascículos de aparición semanal, encuadernables en ocho volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-7598-034-7 (tomo 6)
84-85822-82-X (obra completa)
Depósito Legal: B. 52/1984

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 228505

Impreso en España-Printed in Spain-Mayo 1985

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, 28034 Madrid.
Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.° 52-03, Bogotá D.E.
Distribuye para México: Distribuidora Intermex, S.A., Lucio blanco, n.° 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.
Distribuye para Venezuela: Distribuidora Continental, S.A., Edificio Bloque Dearmas, final Avda. San Martín con final Avda. La Paz, Caracas 1010.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)

Las condiciones de suscripción a la obra completa (96 fascículos más las tapas, guardas y transferibles para la confección de los 8 volúmenes) son las siguientes:

a) Un pago único anticipado de 19 425 ptas. o bien 8 pagos trimestrales anticipados y consecutivos de 2 429 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 6.850.277 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Editorial Delta, S.A. (Paseo de Gracia, 88, 5.°, 08008 Barcelona), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Editorial Delta, S.A., en la forma establecida en el apartado b).

Para cualquier aclaración, telefonear al (93) 215 75 21.

No se efectúan envíos contra reembolso.





Mensaje recibido

En este capítulo analizaremos los protocolos que se precisan para que dos ordenadores se comuniquen a través de un modem

Steve Cross

El término "dúplex" nos indica si un sistema puede o no transmitir y recibir datos al mismo tiempo. Un sistema dúplex total *(full-duplex)* permite la transferencia bidireccional de datos, mientras que los sistemas medio dúplex *(half-duplex)* no lo permiten. Esto se puede comparar con la diferencia que existe entre hablar por teléfono, con dos interlocutores que pueden hablar a la vez, y utilizar un radio-teléfono, en el cual la pulsación del botón de transmisión impide automáticamente la recepción.

La principal ventaja de un sistema dúplex-total es su capacidad para interrumpir al ordenador transmisor. Si, por ejemplo, se está transmitiendo un menú largo y usted ya sabe que desea la opción 1, puede pulsar "1" y el sistema actuará de inmediato sobre esa entrada. En un sistema medio dúplex, tendría que esperar hasta que se transmitiera el menú completo antes de efectuar su selección.

El dúplex total se obtiene mediante el empleo de frecuencias distintas para cada máquina. El ordenador que hace la llamada utilizada una frecuencia que se conoce como la «de origen», mientras que el ordenador anfitrión utiliza la frecuencia «de respuesta».

Protocolos de terminal

Cuando un ordenador transmite datos a otro a través de una línea telefónica, inicialmente no sabe nada acerca del ordenador del extremo receptor. El terminal puede ser un Spectrum con una visualización de 32 columnas o un teletipo de 132 columnas. Existe la posibilidad de que tenga o no color. De que soporte o no caracteres en minúscula. En resumen, puede tratarse de cualquier máquina, desde un micro hasta un ordenador central.

Existen tres formas de abordar este problema. La primera es el enfoque seguido por el lenguaje BASICODE, que adopta el mínimo denominador común y transmite sólo aquellos formatos que puedan ser manipulados por el terminal equipado más elementalmente. La segunda consiste en solicitar información respecto a las capacidades del terminal dado y luego modificar la salida para adecuarla a ellas. El otro enfoque es aquel en el cual el sistema da por sentado que ha accedido a un determinado terminal, o a un tipo de terminal, y deja que el software del usuario haga frente a los datos transmitidos.

El primer enfoque se aplica muy raramente, dado que el mínimo denominador común entre terminales es el teletipo estándar. Éste sólo posee un juego de caracteres en mayúscula, carece de formateado de impresión, de color y de gráficos y es de poca velocidad. El segundo procedimiento es más común: es el que utilizan la mayoría de los tablones de anuncios. El tercer método es el que suelen emplear los sistemas comerciales y universitarios, y se

Fuera de lugar
En los anuncios publicitarios se suelen mostrar micros portátiles y modems acústicos que operan a pilas funcionando en toda clase de lugares; pero existen algunas situaciones en que *no* pueden ser utilizados. Algunos ejemplos en este sentido son los teléfonos públicos comunes, las centralitas telefónicas manuales (que se emplean en muchos hoteles) y en las líneas de baja calidad y con muchas interferencias, problema éste que afecta a numerosas zonas apartadas

Observando el protocolo
Existen tres enfoques principales para los protocolos de comunicaciones. El primero adopta el mínimo denominador común (velocidad de transmisión reducida, pocas columnas de texto, ausencia de color y de gráficos, etc.). En el segundo la máquina anfitriona modifica su salida con el fin de adaptarse a diversos terminales (técnica que aplican la mayoría de los tablones de anuncios). El tercero es para que el terminal se "comporte" como un determinado terminal; esto se consigue mediante una técnica denominada *emulación de terminal*

basa en una técnica de software que se conoce como *emulación de terminal*.

Como su nombre sugiere, la emulación de terminal es simplemente un método para persuadir a un micro de que actúe como un terminal dado. De forma muy simple, el software para emulación de terminal traduce los caracteres de control de terminal entrantes (tales como los que se utilizan para limpiar la pantalla o posicionar el cursor) en instrucciones comprensibles para el ordenador que se está empleando. Del mismo modo, si el anfitrión espera del terminal una secuencia de caracteres de control, el software de emulación se la proporcionará.

Casi todos los sistemas comprenderán un subjuego de caracteres de control ASCII. Algunos de los caracteres más útiles son Control-S (ASCII 19, conocido como "XOFF"), que detiene temporalmente la recepción de datos; Control-Q (ASCII 17, conocido como "XON"), que la restablece; Control-J (ASCII,10, conocido como "LF"), que fuerza un salto de línea sin un retorno de carro, y Control-G (ASCII 7, conocido como "BEL"), que hace sonar la campanilla de la consola. El último carácter puede ser útil si se necesita atraer la atención del operador del sistema del terminal receptor.

Transmisión con un protocolo

En el sistema de protocolo único, la máquina anfitriona ha de restringir sus transmisiones para adaptarse al terminal más burdo. Todos los terminales, no importa cuán sofisticados sean, reciben la misma salida restringida

Transmisión con varios protocolos

La máquina anfitriona adapta su transmisión para ajustarse a las capacidades de cada terminal. Un Spectrum recibirá una salida de 32 columnas, mientras que un BBC recibirá texto en Mode 0, a 80 columnas

Emulación de terminal

Algunos sistemas anfitriones dan por sentado que los usuarios poseen un determinado terminal. Si posee uno diferente, su software habrá de convertir tanto la entrada como la salida para emular las características de dicho terminal

Kevin Jones

El protocolo XModem

Habiendo visto detalladamente cómo se transmiten los datos ASCII, consideremos ahora cómo se comunica el software. La transmisión de software escrito en BASIC es directa. La mayoría de los micros soportan algún medio de convertirlos en su formato de programa comprimido a forma ASCII; en el BBC utilizamos *SPOOL; en el Commodore 64, LISTamos un archivo en disco o cinta; en el Tandy se emplea la instrucción CSAVE<nombrearchivo>,A; etc. El archivo así decodificado se transmite después y es cambiado nuevamente de formato en el otro extremo.

Sin embargo, los archivos CP/M de instrucciones (.COM) no se pueden convertir a forma ASCII, y cualquier intento por transmitir uno resulta un fracaso. Por este motivo se desarrolló un protocolo denominado *XModem*. El XModem, que es una característica de algunos software de comunicaciones, simplemente lee un archivo CP/M del disco y lo transmite en un formato binario estándar. Por supuesto, el terminal receptor debe soportar XModem.

En un futuro capítulo estudiaremos con mayor detalle el empleo de las comunicaciones por ordenador. De momento veamos someramente la gama de actividades en que se puede participar al disponer de un modem.

Correo electrónico es el nombre que identifica a los sistemas en los que los usuarios pueden intercambiar mensajes privados mediante la transmisión de los mismos a un ordenador central, donde se los almacena hasta que el destinario conecta con el sistema y los recupera.

A la mayor parte de los usuarios de micros personales les interesan dos aplicaciones determinadas. La primera es el intercambio de software a través del teléfono. Como ya hemos visto, transmitir programas en BASIC es muy simple y resulta más rápido, más sencillo y más económico que enviar cassettes por correo. Si está escribiendo un programa que le plantea dificultades, puede transmitirle una copia del mismo a sus amigos para ver si pueden ayudarlo. De ser así, ¡ellos le vuelven a transmitir a usted una versión operativa!

El segundo uso de los aficionados son los tablones de anuncios. Éstos permiten que usted le pase información a otros usuarios, deje notas solicitando ayuda técnica, gaste bromas, cargue software de dominio público, practique juegos, etc.

Los tablones de anuncios los llevan aficionados y normalmente no cobran por los mismos ninguna tarifa (si bien algunos tablones pueden cobrarle una tarifa nominal inicial, de alrededor de 250 ptas., con el objeto de cubrir los costos de funcionamiento). Más adelante analizaremos en profundidad los tablones de anuncios. En el próximo capítulo, no obstante, veremos cómo seleccionar modems y software para comunicaciones.



Muchos motores

Le enseñamos cómo controlar varios servomotores simultáneamente, conectándolos al ordenador a través de la puerta para el usuario

Hay ocho líneas de datos que se pueden conectar a motores, si bien la caja buffer que hemos diseñado sólo puede utilizar cuatro. Se podrían emplear las ocho líneas duplicando el sistema de circuito de la caja de salida para las otras cuatro.

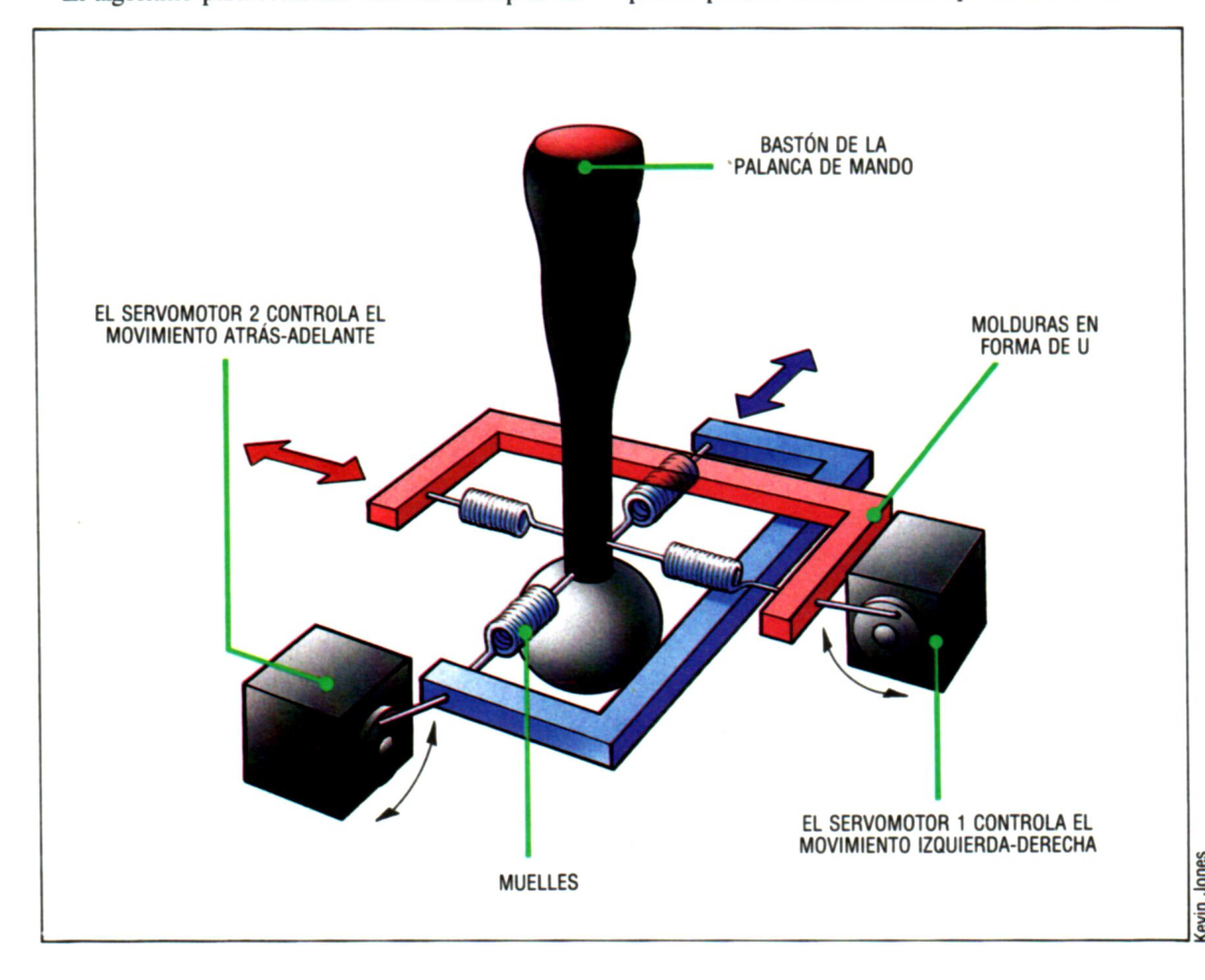
Para controlar simultáneamente ocho motores debe modificarse ligeramente el algoritmo para un solo motor que hemos desarrollado. Los impulsos son iniciados todos juntos, pero el segundo bucle de espera se reemplaza por una tabla de referencia. Se reservan 255 posiciones de memoria para la tabla y se establecen inicialmente en 255 ($FF).

Luego se entran en esta tabla de excepciones (cuando hay un motor apagado). Por ejemplo, si la línea de datos 2 se ha de apagar tras una cuenta de 20, se alterará la vigésima entrada de la tabla, de 11111111 binario ($FF) a 11111011 ($FB). Observe que en el listado en assembly esto se efectúa mediante un AND con el valor que ya esté en la tabla. Después de entradas todas las excepciones en la tabla, se inicia el bucle de espera, pero esta vez cada elemento de la tabla se opera mediante AND con la puerta para el usuario.

El algoritmo para controlar motores múltiples es:

1) Especificar el ángulo de cada motor almacenando los ángulos en ocho bytes (de ANGULO+0 a ANGULO+7).
2) Establecer altos todos los bits de datos de la puerta para el usuario, para iniciar todos los impulsos simultáneamente.
3) Insertar las excepciones en la tabla de referencia.
4) Esperar durante un milisegundo.
5) Cargar en el acumulador el número binario 11111111 ($FF). Luego operar el acumulador mediante AND con cada elemento de la tabla de referencia, de uno en uno. Al encontrar una excepción, se apagará el bit apropiado. Dado que la operación mediante AND continúa hasta el final de la tabla, este bit permanecerá apagado hasta el final.
6) Volver a establecer las excepciones de la tabla otra vez al binario 11111111, a punto para el próximo impulso.

Los listados ofrecidos para el BBC Micro poseen la misma rutina inicial (líneas 10 a 280) en ambos programas. El primer listado es para el control de un solo servomotor conectado a una de las líneas de la puerta para el usuario. El temporizador de eventos



Realimentación por palanca de mando
Las palancas de mando se emplean normalmente para proporcionar información direccional a utilizar por el software. Una posible aplicación para ordenador de los servomotores consiste en emplearlos para permitir que el software controlador realimente con su información el bastón de la palanca de mando. Se utilizan dos servomotores para empujar y tirar del bastón en cada plano horizontal, proporcionando importantes datos de realimentación en los simuladores de vuelo más perfeccionados. Una palanca de mando que "se oponga" bajo la mano en respuesta a un movimiento de control por parte del piloto, mejora la simulación al proporcionar al usuario una información táctil además de visual

Mecanismo de dirección

SERVOMOTOR
EJE
RUEDA
VARILLA DE UNIÓN
PIVOTE O JUNTURA UNIVERSAL

En nuestro robot el control direccional se obtiene mediante el control independiente de dos motores paso a paso bidireccionales. Una posible alternativa a esta disposición sería tener un motor paso a paso para activar el vehículo y un servomotor para dirigirlo. El diagrama muestra un servomotor montado sobre el eje de las ruedas, empujando o tirando de las varillas que conectan las ruedas para dirigir el vehículo. Otra disposición podría ser utilizar un mecanismo de dirección de cremallera y piñón, montando el engranaje del piñón en el husillo del servomotor

se prepara mediante BASIC. Un procedimiento de inicialización ensambla la rutina manejadora de eventos antes de que el programa principal la ejecute permitiendo el evento 5.

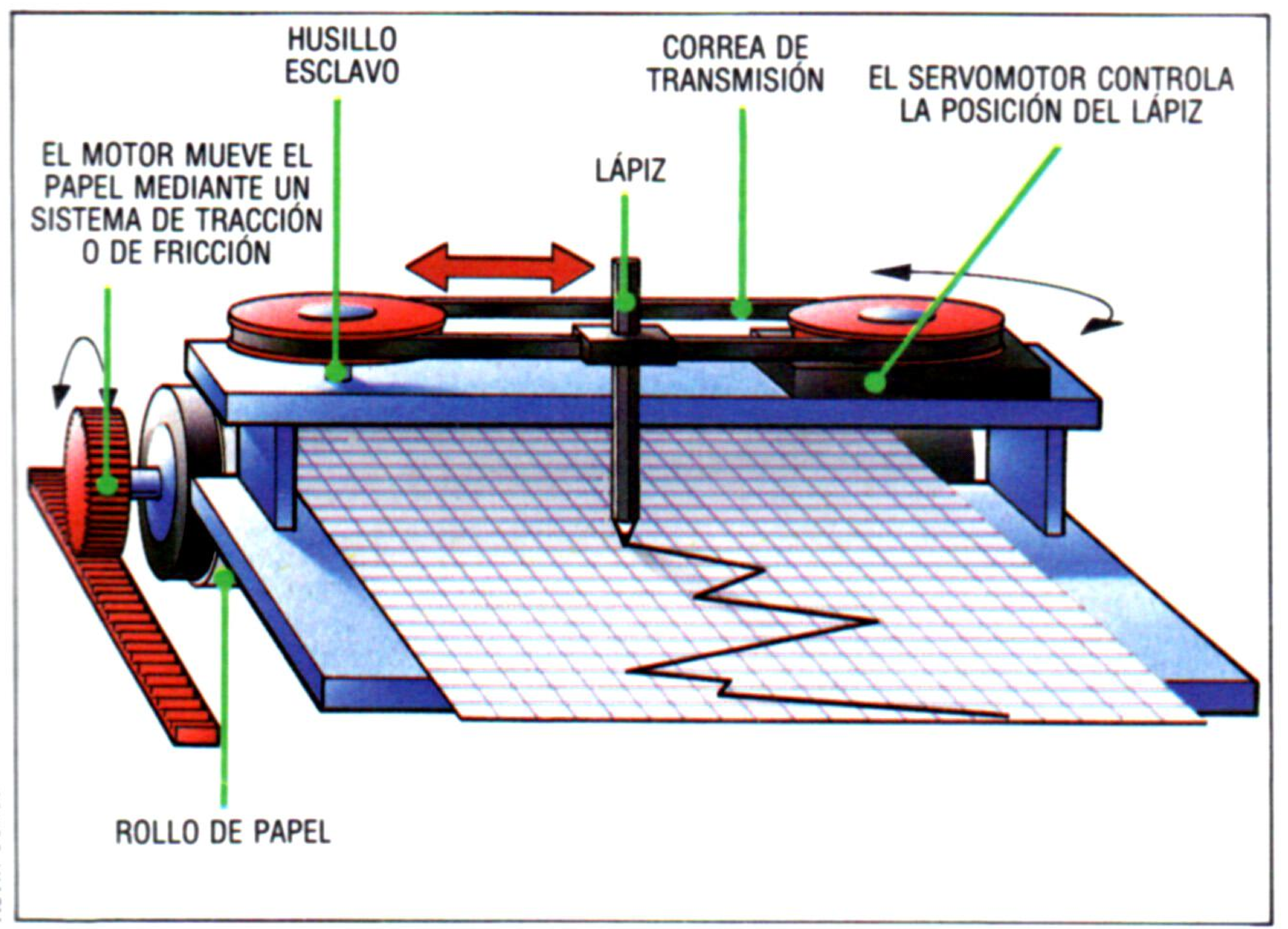
En el BBC Micro, el sistema operativo incluye un temporizador para el usuario en centésimas de segundos. Estableciéndolo en 2 centisegundos (dos interrupciones de 10 milisegundos cada una) y utilizando luego el vector de "eventos", el procesador saltará al código "impulsador" en el momento adecuado. Puesto que el sistema operativo se diseñó para emplear los eventos, el programa sólo ha de retornar (RTS) desde la subrutina, y no usar RTI.

El segundo listado, para el control de múltiples servomotores, utiliza primero un bucle en BASIC para inicializar la tabla de referencia con valores $FF. Si cada elemento de la tabla saliera de uno en uno hacia la puerta para el usuario, todos los impulsos continuarían durante dos milisegundos. Sin embargo, se pueden hacer excepciones y apagar cada motor por turno. Las excepciones se insertan en la tabla empezando por el motor 7, mediante el empleo de un desplazamiento (en el registro X) proporcional a la longitud del impulso. Luego la tabla sale hacia la puerta para el usuario mediante el direccionamiento indirecto de cada elemento de uno en uno, esta vez utilizando el direccionamiento indirecto postindexado (donde el procesador le suma el valor del registro Y a la dirección hallada en un vector de página cero).

Los patrones de bits (excepciones) que deben enviarse a la puerta para el usuario para controlar los motores se producen del siguiente modo. Se requiere el binario 01111111 para apagar el motor 7; el binario 10111111 para el motor 6; el 11011111 para el motor 5, etc. Éstos se generan cargando el registro A con $FF y limpiando el flag o indicador de arrastre. Luego, como la rutina trata cada excepción de una en una, estos bits se desplazan una posición hacia la derecha. En el primer desplazamiento, el bit de arrastre se desplaza al bit 7, el bit 7 se desplaza al bit 6, y así sucesivamente, con el bit 0 reemplazando al bit de arrastre. El patrón de bits requerido para apagar cada motor se guarda temporalmente en la pila. Cada entrada de la tabla se opera mediante AND con la entrada anterior (cuando ésta sale) para asegurar que una vez apagado un impulso éste permanezca apagado.

Habiendo generado el código máquina, el evento 5 se limpia para la acción. Por consiguiente, pulsando la tecla Shift junto con una tecla numérica del 1 al 8 se selecciona uno de los motores, mientras que pulsando las teclas del 1 al 9 los motores se colocan en posición.

Trazador de gráficos
Se puede diseñar un trazador de gráficos utilizando un servomotor para mover el lápiz hacia adelante y hacia atrás, mientras un motor paso a paso va moviendo el papel por debajo de aquél. El movimiento angular del husillo del servomotor se traduce en el movimiento lineal hacia adelante y atrás del lápiz gracias a una correa de transmisión. El movimiento del lápiz se corresponde con los cambios de una variable del eje vertical (p. ej., temperatura o presión barométrica); la alimentación continua del papel depende a menudo del transcurso del tiempo



Kevin Jones

Utilización de los listados

Para usar los listados para el BBC, simplemente éntrelos, guárdelos y luego ejecútelos. Tanto el listado de un solo servomotor como el de múltiples servomotores se ejecutan con la rutina inicial común (de la línea 10 a la 750).

Para el Commodore 64, el segundo algoritmo utiliza las mismas teclas de instrucciones que el listado para el BBC. El programa de llamada en BASIC permite establecer de forma independiente la posición de cada motor: la tecla Shift y una tecla numérica del 1 al 8 seleccionan el motor deseado, y una tecla del 1 al 9 define la posición requerida.

Si posee un ensamblador, entonces digite el listado fuente y ensámblelo en un archivo objeto que subsiguientemente se pueda cargar mediante el programa de llamada en BASIC. De no ser así, digite el cargador en BASIC para el lenguaje máquina y ejecútelo. Digite NEW antes de cargar y ejecutar el programa de llamada en BASIC. Si utiliza el cargador en BASIC, puede omitir las líneas 30 y 40.



Commodore 64: control de servomotores múltiples

Código fuente

```
1000 ;+++++++++++++++++
1010 ;+++++++++++++++++
1020 ;++                 ++
1030 ;++   CONTROL MULTIPLES   ++
1040 ;++   SERVOMOTORES CBM    ++
1050 ;++                 ++
1060 ;+++++++++++++++++
1070 ;+++++++++++++++++
1080 ;
1090 PUERTA=56577      ;REGISTRO DATOS PUERTA USUARIO
1100 ANGULO=1228       ;POSICION VALOR ANGULO
1110 CPAG=$FB          ;PUNTERO PAGINA 0 A TABLA

1120 ;
1130 *=$0334
1140 ;
1150        SEI         ; INTERRUPCIONES DESACTIVADAS
1160        LDA $0314   ; VECTOR IRQ EXISTENTE
1170        LDX $03C4
1180        STA $03C4
1190        STX $0314
1200        LDA $0315
1210        LDX $03C5
1220        STA $03C5
1230        STX $0315
1240 ;
1250 ;++++ INICIALIZAR TABLA ++++
1260 ;
1270        LDA #$FF
1280        LDY #$00
1290 TABLA
1300        STA (CPAG),Y
1310        DEY
1320        BNE TABLA
1330        CLI              ; INTERRUPCIONES ACTIVADAS
1340        RTS
1350 ;
1360 ;++++ MANEJADOR EVENTOS ++++
1370 ;
1380        PHP
1390        PHA              ; GUARDAR REGISTROS
1400        TYA              ; EN LA PILA
1410        PHA
1420        TXA
1430        PHA
1440 ;
1450 ;++ EMPEZAR IMPULSO, PARA ALGUNOS MOTORES
1460 ; SE PODRIA COMENZAR ANTES DE LLENAR LA
1470 ;TABLA Y REDUCIR ASI EL BUCLE DE ESPERA DE ABAJO ++
1480 ;
1490        LDA #$FF
1500        STA PUERTA
1510 ;++ LLENAR TABLA CON EXCEPCIONES ++
1520        LDX #$07
1530        LDA #$FF
1540        CLC
1550 EXCEPT
1560        ROR A
1570        PHA              ; PATRON DE BITS
1580        LDY ANGULO,X     ; TOMAR DESPLAZ. MOTOR X
1590        AND (CPAG),Y     ; CONSERVAR PATRON EXISTENTE
1600        STA (CPAG),Y     ; PERO MODIFICADO PARA MOTOR X
1610        PLA
1620        DEX
1630        BPL EXCEPT
1640 ;++ AHORA LA TABLA ESTA CARGADA ++
1650        LDY #$30
1660 ESPERA
1670        DEY              ; DEJAR PASAR
1680        BNE ESPERA       ; UN POCO DE TIEMPO
1690 ;
1700        LDA #$FF         ; TODOS LOS IMPULSOS ENCEN-
1710        LDY #$00         DIDOS
1720 BUCLE
1730        AND (CPAG),Y     ; PERO OPERAR CON MASCARA
1740        STA PUERTA       CADA ELEMENTO ; DE LA TABLA
1750        INY              POR TURNO
1760        BNE BUCLE
1770 ;
1780        LDX #$07
1790        LDA #$FF
1800 LIMPIA
1810        LDY ANGULO,X     ; BORRAR TODAS LAS EXCEPCIONES
1820        STA (CPAG),Y
1830        DEX
1840        BPL LIMPIA
1850 ;++ AHORA SE DEBEN TERMINAR TODOS LOS IMPULSOS ++
1860        PLA
1870        TAX
1880        PLA              ; RESTAURAR REGISTROS
1890        TAY
1900        PLA
1910        PLP
1920        JMP $EA31
```

Programa cargador en BASIC

```
10 REM ****   CARGADOR EN BASIC PARA   ****
20 REM ****   MULTIPLES SERVOMOTORES   ****
30 :
40 FOR I=820 TO 922
50 READ A: POKE I,A
60 CC=CC+A
70 NEXT I
80 READ CS:IF CS<>CC THEN
   PRINT"ERROR EN SUMA DE
   CONTROL":STOP
100 DATA120,173,20,3,174,196,3,141,196
110 DATA3,142,20,3,173,21,3,174,197,3
120 DATA141,197,3,142,21,3,169,255,160
130 DATA0,145,251,136,208,251,88,96,8
140 DATA72,152,72,138,72,169,255,141,1
150 DATA221,162,7,169,255,24,106,72
160 DATA188,0,48,49,251,145,251,104
170 DATA202,16,243,160,48,136,208,253
180 DATA169,255,160,0,48,251,141,1,221
190 DATA200,208,248,162,7,168,255,188
200 DATA0,48,145,251,202,16,248,104
210 DATA170,104,168,104,40,76,49,234
220 DATA13072:REM"SUMA DE CONTROL"
```

Programa de llamada en BASIC

```
10 REM ****   MULTIPLES SERVOMOTORES   ****
20 :
30 DN=8:REM SI CASSETTE ENTONCES
   DN=1
40 IF A=0 THEN A=1:LOAD"MULTISER V.
   HEX",DN,1
50 POKE 778,88: POKE 779,3:REM
   PUNTERO A MANEJADOR EVENTOS
60 POKE 251,0: POKE 252,49:REM
   ESTABLECER PUNTERO PAGINA CERO
70 RDD=56579: POKE RDD,255:REM
   TODAS SALIDA
80 MC=820: SYS MC
90 :
100 GET K$:IF K$="" THEN 100:REM
    ESPERAR TECLA
110 REM ** ALTERAR POSICION MOTOR **
115 AK=ASC(K$)
120 IF AK>48 AND AK<58 THEN POKE
    12288+SERVO,VAL(K$)*20
130 IF AK>32 AND AK<40 THEN
    SERVO=ASC(K$)-35
140 IF K$<>"E" THEN 100:REM "E" PARA
    SALIR
```

Control de un solo servomotor

```
755 REM ****************************************
756 REM *          Ensamblar el codigo maquina
757 REM ****************************************
10000 DEF PROCinicial
10010 DIM espacio% 200
10020 FOR C=0 TO 2 STEP 2
10030    puertab=&FE60
10040    P%=espacio%
10050    angulo=P%
10060    P%=P%+1
10070    [OPT C
10080    .manejador eventos
10090    /guardar registros primero
10100    PHP:PHA:TYA:PHA:TXA:PHA
10110    LDA £&04
10120    LDX £xreloj
10130    LDY £yreloj
10140    JSR &FFF1 reinicializar reloj
10150    LDA £&FF
10160    STA puertab
10170    /esperar aprox. 1mseg
10180    LDY £&FF
10190    .BUCLE DEY
10200          BNE BUCLE
10210    /y contar impulso
10220    LDY angulo
10230    .BUCLE1 DEY
10240          BNE BUCLE1
10250    /detener todos los impulsos de salida
10260    LDA £&0
10270    STA puertab
10280    PLA:TAX:PLA:TAY:PLA:PLP
10290    RTS
10300    ]
10310    NEXT C
10320 REM apuntar al manejador eventos
10330 !&220=manejador eventos DR (!&220 AND &FFFF0000)
10340 ENDPROC
```

Listados para el BBC Micro

Rutina inicial común

```
10 MODE 0
15 REM ****************************************
16 REM            Preparar el temporizador etc.
17 REM ****************************************
20 osbyte=&FFF4
30 A%=&97 :X%=&62 :Y%=&FF
40 CALL osbyte: REM preparar puerta B para salida
50 CLS
60 DIM p%(8)
70 DIM reloj% 12, leer% 12
80 xreloj=reloj% MOD 256
90 yreloj=reloj% DIV 256
100 xleer=leer% MOD 256
110 yleer=leer% DIV 256
120 PROCinicial
130 FOR I%=angulo TO angulo+8:angulo?I%=128:NEXT
140 t=.02 :REM seg entre impulsos
150 hora%=&FFFFFFFF -(t*100)+1
160 reloj%?4=&FF      :REM cargar byte mas alto
170 !reloj%=hora%      :REM establecer reloj, permitir eventos
180 +FX14,5
190 A%=4:X%=xreloj  :Y%=yreloj  :CALL &FFF1
195 REM ****************************************
196 REM              El controlador BASIC
197 REM ****************************************
200 CLS
210 PRINT"PULSAR SHIFT+NUMERO para seleccionar motor"
220 PRINT"PULSAR NUMERO para seleccionar angulo"
225 motor=1
230 REPEAT
240    A=GET
250    IF A>&2F AND A<&3A THEN
       a=(A-&30)*10*(225/90):angulo?(motor-1)=a:PRINTTAB
       (10,motor);"motor, "motor "angulo "angulo "a+(90/255)
260    IF A>&20 AND A<&2A THEN motor=A-&20
270    UNTIL 0
280 END
750
```

Control de múltiples servomotores

```
755 REM ****************************************
756 REM           Ensamblar el codigo maquina
757 REM ****************************************
760 DEF PROCinicial
770 DIM espacio% 600
780 FOR C=0 TO 3 STEP 3
790    paginacero=&70      :REM libre para usuarios
800    puertab=&FE60 :osword=&FFF1
810    P%=espacio%
820    angulo=P% :P%=P%+8   :REM potencialmente 8 motores
830    tabla=P% :P%=P%+256 :REM 256 longitudes de impulso
       posibles
835    FORI%=tabla TO tabla+&100:?100:?I%=&FF:NEXT
840    lowtabla%=tabla MOD 256
850    hightabla%=tabla DIV 256
860    ?paginacero=lowtabla% :paginacero?1=hightabla%
870    [OPT C
880    .manejador eventos
890    PHP:PHA:TYA:PHA:TXA:PHA
900    LDA £&04
910    LDX £xreloj
920    LDY £yreloj
930    JSR osword
990     /Comenzar impulso, para algunos motores seria posible
        comenzar de abajo
1000    /antes de llenar la tabla y reducir asi el bucle de esperar de abajo
1010   LDA £&FF :STA puertab
1020    /llenar tabla con excepciones
1030   LDX £&7 :LDA £&FF :CLC /preparar patron bits
1040   .excepciones
1050      ROR A :PHA /patron de bits
1060      LDY angulo,X /tomar desplazamiento correspondiente al angulo
          del motor X
1070      AND (paginacero),Y /conservar patron bits existente
1080      STA (paginacero),Y /pero modificado para motor X
1090      PLA :DEX
1100      BPL excepciones
1110   /ahora la tabla esta cargada, dejar pasar un poco de tiempo
1120   LDY £&60
1130   .esperar DEY
1140          BNE esperar
1150   LDA £&FF /todos los impulsos encendidos
1160   LDY £&0
1170   .bucle AND (paginacero), Y /pero enmascarar cada elemento
1180         STA puertab /de la tabla por turno
1190         INY
1200         BNE bucle
1201   LDX £&7 :LDA £&FF
1203   .limpiar
1205      LDY angulo,X /volver a borrar todas las excepciones
1207      STA (paginacero),Y
1208   DEX
1209   BPL limpiar
1210    /ahora todos los impulsos deben haber terminado
1220   PLA:TAX:PLA:TAY:PLA:PLP
1230   RTS
1240   ]
1250   NEXT C
1260 !&220=manejador eventos OR (!&220 AND &FFFF0000)
1270 ENDPROC
```

Aprender jugando

Esta vez revisaremos diversos programas educativos destinados a los niños más pequeños

Story machine (La máquina de las historias), producido por Spinnaker, está diseñado para niños de cinco a nueve años de edad. A través de él se pretende enseñar las reglas de la sintaxis, mejorar la ortografía y estimular la escritura expresiva.

El programa contiene un diccionario de 52 palabras, de las cuales cinco son nombres propios: los nombres de un niño, una niña, un gato, un perro y una criatura denominada *bumpus*, todos los cuales deben ser especificados por el usuario al comienzo del programa.

A partir de este diccionario el niño construye frases simples que luego son representadas por los personajes en la pantalla. Por ejemplo, la frase "Mauricio besa flores" visualizará un personaje que represente a un niño (al cual el usuario habrá llamado Mauricio), unos caracteres con forma de flores y, cuando éstos se acerquen lo suficiente, aparecerán corazones intermitentes indicando los besos.

Si una palabra está mal escrita o se la emplea de forma incorrecta, el ordenador se negará a aceptarla, visualizando un mensaje en que explica lo que está mal y en que pide que la palabra se modifique o se sustituya. Al final de cada frase (es decir, después de cada punto) el programa intentará ponerla en escena. Cuando se ha construido una historia, hay una opción para representarla completa.

Story machine (La máquina de las historias)



Si el usuario se aburre de escribir sus propias historias, *Story machine* ofrece otras dos opciones: puede compartir la escritura de la historia con el programa (turnándose para añadir palabras) o dejar que el ordenador haga todo el trabajo y que escriba él la historia. Esto es factible porque el diccionario está dividido por elementos de la frase (sustantivos, verbos, preposiciones, etc.) y posee un algoritmo gramatical estricto para generar sus propias frases.

La mayoría de las palabras del diccionario, así como las estructuras oracionales utilizadas por el programa, son muy sencillas. Los verbos, en particular, describen acciones que se pueden describir fácilmente, tales como "salta" y "va". Es interesante el hecho de que se haya incluido un verbo imaginario, *zot*, el cual parece significar "pegar" o "golpear". De ser así, ello estaría en desacuerdo con la regla mantenida por Spinnaker, de que está mal herir a otras personas o a animales, y por lo tanto el programa no admitirá ninguna acción que implique un daño físico a otros.

El programa posee dos reglas gramaticales fundamentales: una oración debe contener un sujeto, un verbo y un objeto, en el orden correcto; y las formas en plural y en singular deben ser coherentes. Más allá de estas reglas, la sintaxis es menos satisfactoria debido a las inmensas variaciones del uso del idioma inglés. Por ejemplo, el ordenador aceptará la frase *"Houses eat rocks"* (Casas comen rocas) pero no aceptará la frase *"Girls eat rocks"* (Chicas comen rocas) sin el artículo definido antes de la palabra "chicas".

El programa parece incluso quebrantar sus propias reglas. Después de la frase *"Bumpuses eat rocks"* (Bumpuses comen rocas), el programa generó la oración *"They walk to its fence"* (caminan hacia su verja). Normalmente, al encontrarse con la palabra *"its"* (su), el ordenador generaría el mensaje *"Whose?"* (¿De quién?) y solicitaría que se cambiase la palabra. El programa también ha hecho caso omiso de su regla de no mezclar formas en singular y plural. Éstos se podrían considerar errores menores, pero un programa que se vende como educativo debería al menos ser coherente y *debe* ser siempre correcto.

Kids on keys (Niños a las teclas) es otro producto de Spinnaker, dirigido a niños de tres a nueve años. El programa tiene como objetivo enseñar al usuario a identificar palabras, letras y números. En el paquete hay tres juegos diferentes, cada uno de los cuales posee varios niveles de dificultad. En el primer juego, una letra se desplaza hacia abajo de la pantalla y el niño debe pulsar la tecla correcta antes de que la letra llegue al límite inferior. Al cabo de 15 letras, aparece flotando hacia abajo un globo que contiene una palabra. Ésta también debe digitarse correctamente antes de que el globo llegue hasta la parte inferior de la visualización. El juego, por lo menos, consigue que el niño disfrute mientras se familiariza con el trazado del teclado.

En el segundo juego se van desplazando hacia abajo de la pantalla una serie de imágenes (sprites ampliados) y el niño debe digitar el nombre del objeto antes de que lleguen a la parte inferior. Después de esto, hay una ronda extra en la cual descienden los mismos objetos, pero en este caso faltándoles dos cuartos de la imagen, lo que hace que resulte más difícil reconocerlos. En este punto, el

Kids on keys (Niños a las teclas)



juego tiende a convertirse en una prueba de reflejos, mientras el jugador busca frenéticamente las teclas para digitar las letras correctas. Un importantísimo inconveniente del programa es la ausencia de una facilidad para borrar, con lo cual la pulsación errada de una tecla se vuelve sumamente frustrante.

Quizá el problema más serio del juego sea que algunos de los sprites están mal dibujados y es difícil identificarlos. Por ejemplo, si un niño decide que un determinado dibujo es un hombre, podría digitar "hombre" de forma correcta, sólo para descubrir que el sprite continúa su descenso porque el programa considera que es un "oso" o un "niño". El pequeño puede entonces llegar a la conclusión de que su ortografía es incorrecta, porque el programa no le explica el motivo del fallo (puesto que no posee facilidad alguna para analizarlo). Además, una vez que el sprite ha llegado a la parte inferior de la pantalla, el programa no ofrece ninguna indicación acerca de cuál era la respuesta acertada. Un buen programa educativo no sólo le proporcionaría al usuario la posibilidad de intentarlo otra vez, sino que también visualizaría la respuesta correcta después de haber obtenido varias respuestas equivocadas.

En el tercer juego se visualizan en la pantalla cinco figuras y una palabra, y el niño debe encontrar la pareja correcta para la palabra. Este juego tiene menos relación con la velocidad que los dos anteriores, si bien también adolece de la pobreza de los gráficos.

Aprender a leer

Macmillan, la editorial británica de publicaciones educativas, ha desarrollado una serie de programas para el Spectrum bajo el título colectivo de *Learn to read* (Aprender a leer). Estos cinco programas (producidos en colaboración con Sinclair Research) derivan del esquema de lectura *Gay way* de Macmillan, que ha obtenido un éxito enorme; se trata de un curso centrado en el niño que se utiliza ampliamente en las escuelas primarias. Dado que el curso *Gay way* está orientado hacia la enseñanza individual, posee la ventaja de que los usuarios aprenden a leer a su propio ritmo. Una serie de programas para ordenador es una ampliación natural de ello, puesto que utilizar un ordenador es evidentemente una actividad orientada hacia el individuo.

Los cinco paquetes componen un curso preliminar de lectura completo. El primer programa presenta seis animales (*Deb* la rata, *Sam* el zorro, etc.) y éstos se mantienen a lo largo de todo el curso para proporcionar un sentido de continuidad y familiaridad.

Los programas dan por sentado que el niño no posee ningún conocimiento previo de ordenadores ni de lectura. En cada programa la selección de las opciones es directa. Por ejemplo, el primer programa visualiza cinco opciones, y rodeando el nombre de cada una va apareciendo, de uno en uno, un rectángulo intermitente. El usuario espera hasta que el "cursor" rectangular rodee la opción requerida y entonces pulsa cualquier tecla. Para utilizar este menú no se necesita emplear ninguna otra instrucción.

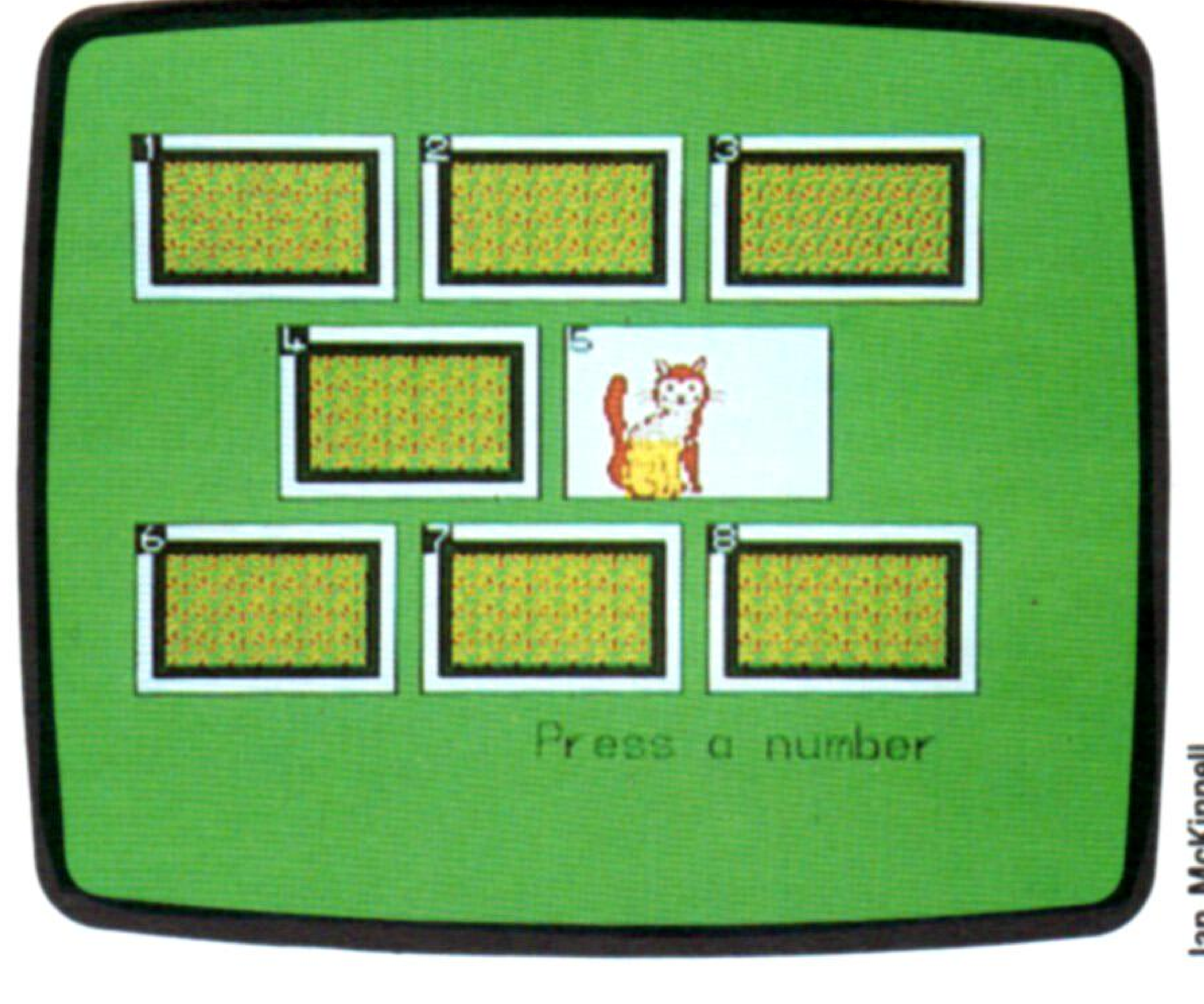
Una vez escogida una opción, el ordenador demuestra lo que se requiere y le solicita luego al niño que dé una respuesta. Si el niño da una respuesta equivocada, se le ofrecerán varias oportunidades más antes de que el ordenador visualice la respuesta correcta.

Los seis animales que aparecen en el programa están dibujados en gráficos de alta resolución, y éstos son muchísimo más atractivos que algunos de los que se incluyen en los otros juegos educativos que hemos analizado. Desde el punto de vista del entretenimiento, quizá el valor de la serie *Learn to read* no sea tan alto como el de muchos otros paquetes, pero su valor educativo parece ser muchísimo mayor, y, a la larga, tal vez mucho más provechoso para el niño.

Learn to read (Aprender a leer)



Learn to read (Aprender a leer)



Ian McKinnell

Gran Premio

Participe en uno de los juegos recreativos que han tenido mayor éxito: "Gran Premio". Esta versión ha sido escrita para los ordenadores Atari



En la pista de carreras, cuatro coches encabezan la competición en una carrera desenfrenada; usted debe evitar tener un accidente mientras trata de recorrer la mayor distancia posible. Utilice la palanca de mando para dirigir su coche y el botón rojo para cambiar de velocidad. ¡Atención!, este programa emplea lenguaje máquina, de modo que tome precauciones antes de lanzarse a competir.

```
0 REM ** GRAN PREMIO, ESCRITO **
1 REM ** POR PAUL DUNNING      **
2 DIM S(4),V$(10),E$(100)
3 GRAPHICS 5:V$="LENT":POKE 752,1
5 GOSUB 1000
6 POKE 765,2:COLOR 2:PLOT 0,0:DRAWTO 79, 0:PLOT
  79,47:DRAWTO 79,30:DRAWTO 0,30:POSITION 0,47:XIO
  18,#6,0,0,"S:"
7 POKE 765,3:COLOR 3:PLOT 79,29:DRAWTO 79,1
  :DRAWTO 0,1:POSITION 0,29:X10 18,#6,0,0,"S:"
8 POKE 712,148
10 PP=PEEK(106)-16:POKE 54279,PP
20 PM=PP*256:POKE 559,62
21 FOR Q=53256 TO 53260:POKE Q,3:POKE
   Q-8,RND(1)*255:NEXT Q
22 POKE 53248,100:POKE 53249,100:POKE 53250,
   100:POKE 53251,50
23 POKE 704,109:POKE 705,89:POKE 706,79: POKE
   707,29:POKE 53277,3:POKE 559,62
24 FOR Q=PM+1024 TO PM+2048:POKE Q,0:NEXT Q
30 RESTORE 210:FOR Q=PM+1144 TO PM+1280:READ
   A:IF A=-1 THEN 32
31 POKE Q,A:NEXT Q
32 RESTORE 210:FOR Q=PM+1360 TO PM+1536:READ
   A:IF A=-1 THEN 34
33 POKE Q,A:NEXT Q
34 RESTORE 210:FOR Q=PM+1576 TO PM+1792:READ
   A:IF A=-1 THEN 36
35 POKE Q,A:NEXT Q
36 RESTORE 210:FOR Q=PM+1842 TO PM+2048:READ
   A+IF A=-1 THEN 40
37 POKE Q,A:NEXT Q
40 POKE 623,1
50 SOUND 0,255,12,5:SOUND 1,145,12,5:SOUND
   2,200,12,5
60 FOR Q=53256 TO 53260:POKE Q,3:POKE
   Q-8,RND(1)*255:NEXT Q
70 POKE 704,109:POKE 705,89:POKE 706,79:POKE
   707,29:POKE 53277,3:POKE 559,62
75 ST=PM+1842:VI=INT(ST/256):LT=ST-256*VI:POKE
   203,LT:POKE 204,VI:POKE 205,10
80 P1=(RND(1)*100)+100:P2=(RND(1)*100)+100:P0=
   (RND(1)*100)+100:P3=50
85 FOR Q=0 TO 3:S(Q)=(RND(1)*2)+1:NEXT Q
87 POKE 53248,P0:POKE 53249,P1:POKE 53250,P2:POKE
   53251,50
88 FOR Q=1 TO 50:X=USR(ADR(E$),7):NEXT Q
90 IF V$="RAP." THEN GOSUB 400:GOTO 120
92 POKE 53278,1
95 P0=P0+S(0):IF P0>255 THEN P0=0
98 X=USR(ADR(E$),STICK((0))
100 P1=P1+S(1):IF P1>255 THEN P1=0
105 X=USR(ADR(E$),STICK(0))
110 P2=P2+S(2):IF P2>255 THEN P2=0
120 KM=KM+0.01
124 X=USR(ADR(E$),STICK(0))
129 IF STR16(0)=0 THEN GOSUB 500
130 POKE 53248,P0:POKE 53249,P1:POKE 53250,P2
135 POKE 656,2:POKE 657,5:? "VELOCID  ";V$;"    ":POKE
    656,2:POKE 657,20:? "KILMS";KM;"  "
140 CP=PEEK(53263):CS=PEEK(53255)
150 IF CP<>0 THEN 600
160 IF CS<>4 THEN 600
190 GOTO 90
210 DATA 64,228,238,238,238,238,78,68,229,245,255,
    255,233,255,255,233,255,255,245,229,68,78,238,238,
    238
215 DATA 238,238,64,-1
300 C=INT(RND(1)*3):S(C)=(RND(1)*2)+1:RETURN
400 P0=P0-(S(0)*2):IF P0<0 THEN P0=255
405 X=USR(ADR(E$),STICK(0))
410 P1=P1-(S(1)*2):IF P1<0 THEN P1=255
415 X=USR(ADR(E$),STICK(0))
420 P2=P2-(S(2)*2):IF P2<0 THEN P2=255
425 X=USR(ADR(E$),STICK(0))
430 KM=KM+0.05:RETURN
500 IF V$="RAP." THEN V$="LENT":SOUND
    0,255,12,10:SOUND 1,245,12,10:SOUND
    2,235,12,10:RETURN
510 V$="RAP.":SOUND 0,100,12,10:SOUND
    1,110,12,10:SOUND 2,90,12,10:RETURN
600 GOTO 640
605 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND 2,0,0,0
610 POKE 53248,100:POKE 53249,100:POKE 53250,100
620 TRAP 620:? "OTRA?";:INPUT V$:IF V$(1,1)="0" THEN
    KM=0:V$="LENT":? CHR$(125):GOTO 24
630 END
640 J=PEEK(203)+256*PEEK(204)
645 FOR Q=1 TO 250:IF PEEK(J+Q)=0 THEN NEXT Q
650 ST=J+Q:VI=INT(ST/256):LT=ST-256*VI:POKE 203,
    LT:POKE 204,VI
655 FOR P=1 TO 60
660 X=USR(1536):SOUND 1,PEEK(53770),120,15
670 POKE 707,PEEK(53770):NEXT P
700 POKE 53251,0:GOTO 605
1000 RESTORE 2000
1010 TRAP 1050:P=0
1020 P=P+1:READ A
1030 E$(P,P)=CHR$(A)
1040 GOTO 1020
1050 RESTORE 3000
1060 TRAP 1090:P=1536
1070 READ A:POKE P,A:P=P+1
1080 GOTO 1070
1090 TRAP 40000
1100 RETURN
2000 DATA 104,104,104,74,72,176,11,160,0,177,203,
     136,145,203,200,200,208,247,104,74,72
2010 DATA  176,11,160,0,177,203,200,145,203,136
2020 DATA 136,208,247,104,74,72,176,7,198,205,165
2030 DATA 205,141,3,208,104,74,176,7,230,205,165,205,
     141,3,208,96,END
3000 DATA 104,160,0,173,10,210,145,203,200,192,25,208,
     246,96,END
```



Llegados de Oriente

El PC-1251 y el PC-1500A, de Sharp, son dos ordenadores de bolsillo ligeros, versátiles y de precio asequible



Potencia de bolsillo
Con RAM CMOS y CPU de 8 bits, teclado QWERTY, BASIC y una gama de periféricos opcionales, estas "calculadoras" Sharp tienen el derecho de ser consideradas microordenadores a pequeña escala y auténticamente portátiles

La estrategia de marketing de la firma japonesa Sharp contrasta notablemente con la de Casio, la otra empresa fabricante de ordenadores de bolsillo. En primer lugar, produce dos máquinas bastante diferentes, mientras que Casio comercializa tres máquinas relativamente similares. Las dos máquinas de Sharp no están diseñadas para competir entre sí: el PC-1251 es el ordenador de bolsillo más pequeño que existe actualmente, y el PC-1500A es considerablemente más grande y más potente.

El Sharp PC-1251 pesa apenas 115 g y mide 135 × 70 × 10 mm. El tamaño del teclado es aun inferior a la mitad de uno estándar y la anchura de las teclas es de sólo 4 mm. Son lo suficientemente grandes, eso sí, para asegurar que, siempre que se tenga cuidado, al pulsar una tecla no se pulsen también las cuatro que la rodean. Al lado del teclado alfabético hay uno numérico con teclas más grandes, que permite utilizar el ordenador como una calculadora de bolsillo normal.

El PC-1251 posee una visualización en cristal líquido de 24 caracteres de anchura. Ésta se puede regular para diversos ángulos de visión y condiciones luminosas mediante una pequeña rueda situada en uno de los extremos del ordenador. A la derecha de la visualización hay un interruptor Mode Selector (selector de modalidad). Hay tres modalidades operativas: una permite definir la función de ciertas teclas (RSV); una segunda modalidad facilita la programación (PRO), y la tercera permite ejecutar un programa en BASIC o utilizar la máquina como calculadora (RUN). Este interruptor también se emplea para apagar la máquina.

A pesar de no responder del todo al estándar de los micros personales comunes, la versión de BASIC utilizada es buena tratándose de una máquina tan pequeña. Posee instrucciones de las que carecen algunos de los ordenadores de bolsillo de Casio, como ASC y CHR$; pero, al igual que éstos, carece de la opción ELSE para la instrucción IF...THEN. Al igual que los ordenadores Casio más pequeños, el BASIC del PC-1251 utiliza los mismos nombres de una sola letra para series y variables. Ello significa que si se empleara la variable A para retener un número, no se puede utilizar la variable en serie A$. Del mismo modo, algunas matrices podrían machacar las mismas zonas de memoria.

Esta versión de BASIC sólo produce nueve mensajes de error y todos letras únicas, lo cual carece prácticamente de utilidad para atrapar errores en los programas. Como opción adicional, la instrucción PASS permite proteger los programas mediante una palabra clave. En el BASIC del PC-1251 los números de línea están limitados a la gama del 1 al 999. En la modalidad de entrada de programas, dos teclas de cursor le permiten al programador ir desplazando las líneas del programa hacia arriba y hacia abajo. En todas las modalidades, otros dos cursores permiten el desplazamiento lateral.

Puesto que es escaso el software disponible para la máquina, la mayoría de los usuarios habrán de escribir sus programas. En este sentido, el manual resulta de ayuda en dos sentidos. En primer lugar, ofrece una guía de BASIC buena y de fácil comprensión, aunque no incluye enseñanza para principiantes. En segundo lugar, contiene los listados de nueve programas cortos, no todos los cuales son aplicaciones matemáticas (hallar raíces, desviación estándar, etc.); entre los otros programas incluidos hay uno para "practicar mecanografía" y un juego de "aterrizaje suave". Además, Sharp ofrece tres cintas que contienen una selección de programas.

Mientras que los ordenadores Casio pueden tener hasta 10 programas simultáneamente en la memoria, el PC-1251 se limitada a uno cada vez. Sin embargo, se puede utilizar un programa compuesto por varios subprogramas, separados cada uno por sentencias END. Un programa se puede de esta manera ejecutar mediante el empleo de GOTO con el número de línea apropiado. Es muy práctico tener varios programas en la memoria al mismo

tiempo, ya que no existe ninguna forma de cargar programas como no sea digitándolos cada vez que se los necesita. Además, el ordenador posee sólo 4 K de memoria, de modo que hay poco lugar para algo más que unos pocos programas modestos en BASIC. Ello significa que, casi para cualquier uso serio, son esenciales la impresora y la cassette opcionales.

Sharp PC-1500A

El otro ordenador portátil de Sharp, el PC-1500A, es un desarrollo de un modelo anterior (el PC-1500) y está diseñado con el objetivo puesto en un usuario más serio. Mide 195 × 85 × 25 mm y pesa 375 g, con lo cual su peso es tres veces mayor que el del PC-1251.

El PC-1500A posee un teclado mejor que el de su hermano y una visualización ligeramente más grande. La anchura de la LCD se amplía mínimamente (26 caracteres en lugar de 24) y en la máquina más grande ha desaparecido la facilidad del PC-1251 para regular la pantalla.

La versión de BASIC, sin embargo, constituye una gran mejora respecto a la otra. Las variables pueden tener nombres de dos letras y se puede utilizar el mismo nombre para variables en serie y numéricas sin producir ninguna confusión. Asimismo, el BASIC soporta algunas útiles facilidades para gráficos y sonido. En la LCD se pueden dibujar patrones con una resolución de 7 filas por 156 columnas, mediante el empleo de la instrucción GPRINT para definir cada una de las columnas de siete puntos. La instrucción BEEP de la máquina permite controlar el tono y la duración de las notas, que son moderadamente fuertes.

El ordenador tiene incorporados un calendario y un reloj, a los cuales se puede acceder mediante la variable TIME. El ordenador mantiene la hora aun cuando esté apagado, de modo que una vez puesto en hora se mantiene haciendo tictac. Esta facilidad sería una adición muy útil para todos los micros personales. El PC-1500A posee en su BASIC otras varias instrucciones que lo colocan en posición ventajosa respecto a muchos micros. Éstas incluyen una instrucción ON ERROR GOTO y facilidades de rastreo (TRON y TROFF). Posee la generosa cifra de 39 mensajes de error para el ordenador estándar; hay otros 16 disponibles para instrucciones utilizadas sólo con unidades accesorias. No obstante, al igual que el PC-1251, estos mensajes de error se visualizan en forma de números, por lo que podrían mejorarse con texto.

La fila superior de las teclas alfabéticas tiene programadas palabras clave del BASIC, y para identificar las 10 instrucciones se les puede instalar encima una plantilla plástica. El motivo por el cual Sharp decidió no imprimir estas instrucciones directamente en la carcasa es todo un misterio: es muy fácil que la plantilla se pierda. Las seis teclas de encima del teclado principal pueden tener programadas hasta 18 funciones.

Los usuarios del PC-1500A tendrán que escribir la mayoría de su software: son pocas las empresas que producen programas para la máquina, y la propia Sharp vende sólo una cinta de programas seleccionados. Con la máquina se suministra un libro con los listados de 53 programas, con diversas aplicaciones para cinco áreas principales: matemáticas, estadística, electricidad, oficina y juegos. Estos programas se escribieron originalmente para el PC-1500, pero todos ellos funcionan a la perfección en el PC-1500A porque la única diferencia entre los dos modelos es la cantidad de memoria RAM disponible. El modelo anterior tenía 3,5 K de memoria, mientras que el PC-1500A posee 8,5 K.

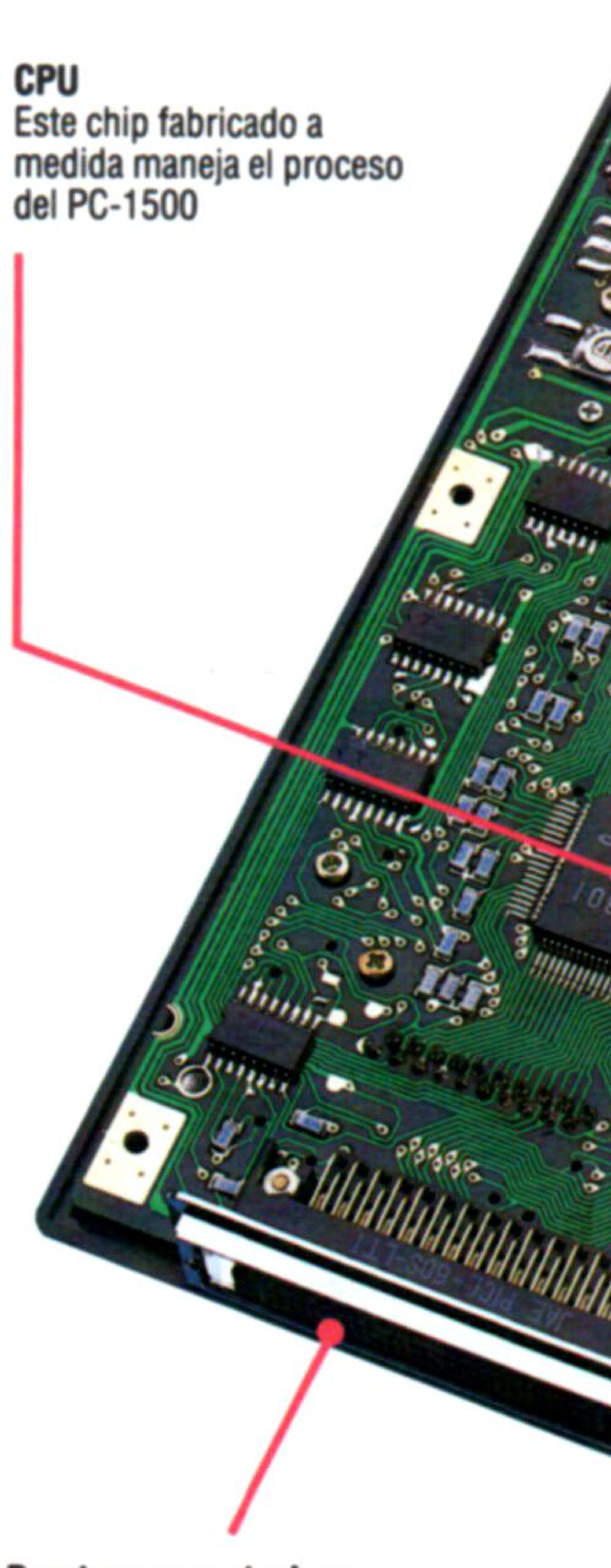
Mediante una ranura para cartuchos situada en la cara inferior de la máquina se puede incrementar la cantidad de memoria disponible. Se ofrecen cuatro cartuchos de memoria y todos ellos son bastante caros. Los paquetes de memoria estándares de 4 y 8 K conservan su contenido sólo mientras están en la máquina. Otros dos cartuchos poseen pilas, de modo que conservan su contenido aun cuando se los extraiga del ordenador. Éstos poseen una capacidad de 8 y 16 K, respectivamente.

La unidad interface para impresora/plotter y cassette representa una mejor relación precio/prestaciones. La interface para cassette permite salvar y cargar programas con una grabadora de cassette común, y Sharp ofrece su propia grabadora de cassette compatible. La parte impresora/plotter de la unidad utiliza cuatro lápices de punta esférica para dibujar letras y gráficos de buena calidad a cuatro colores. Es casi idéntica a otras muchas impresoras-/plotter que existen en el mercado de ordenadores personales, pero el papel que utiliza es de sólo 57 mm de ancho, lo que constituye una gran limitación.

El BASIC incluye un juego completo de instrucciones para utilizar la impresora/plotter. Éstas son: CSIZE para producir letras de diferente tamaño, ROTATE para imprimir caracteres de forma apaisada o invertida, COLOR para seleccionar el color del lápiz, LF para desplazar el papel hacia arriba y hacia abajo, LPRINT para imprimir texto, LCURSOR y GLCURSOR para desplazar el lápiz en modalidades texto y gráficos, SORGN para establecer el margen, y LINE y RLINE para dibujar una línea entre dos puntos empleando coordenadas absolutas y relativas, respectivamente. La unidad para impresora/plotter y cassette se suministra en su propia carcasa, junto con accesorios tales como un transformador de corriente (con un cable para un enchufe común) para alimentar sus pilas recargables.

Para el PC-1500A se producen otros dos accesorios importantes. Uno de ellos es una interface Centronics y RS232, que permite a la máquina comunicarse con impresoras y ordenadores de tamaño natural. Al otro periférico se lo denomina *tablero de software*. Este dispositivo es un gran teclado sensible al tacto que posee 140 teclas definibles que se pueden programar para llevar a cabo tareas utilizadas comúnmente (como calcular totales de forma automática en aplicaciones de hoja electrónica). Dado que el tablero de software es caro y para su operación requiere la interface, el precio de esta facilidad de ampliación es más bien prohibitivo.

A pesar de que el PC-1500A se puede ampliar convirtiéndolo en un sistema potente con muchas facilidades de calidad notable, hay muchos sistemas de micros de precio similar que son más versátiles y (lo más importante) están apoyados por una gama de software más amplia. Con su intento por encabezar la liga de los ordenadores de peso ligero, el PC-1500A corre el peligro de verse aventajado con holgura por muchísimos pesos medios más potentes.



Caja de las pilas
Contiene las cuatro pilas necesarias para que funcione el ordenador

CPU
Este chip fabricado a medida maneja el proceso del PC-1500

Puerta para cartuchos
Esta interface permite instalar en el PC-1500 paquetes de RAM adicional. También se puede conectar a la interface para impresora/cassette o a una interface RS232





Enchufe red
Como alternativa al empleo de pilas, el ordenador puede funcionar con la corriente de la red mediante un transformador adecuado

ROM de BASIC
Este chip contiene el BASIC Microsoft que utiliza el ordenador

Extra Sharp
El Sharp PC-1251 se instala en la unidad impresora/microcassette CE-125. Ésta contiene una impresora térmica (24 caracteres por línea) y una grabadora de microcassette. La unidad mide 205 × 149 × 23 mm.
El Sharp PC-1500 es el centro de una familia ampliada de equipos Sharp, y conectable en interface a través de la interface RS232C/en paralelo CE-158 a prácticamente cualquier sistema de micro u ordenador central. La interface CE-150, para impresora a color/cassette, es un plotter X-Y a cuatro colores con nueve tamaños de caracteres y opción para gráficos. La interface para cassette permite la conexión con dos grabadoras de cassette.
Los módulos de memoria CE-151, CE-155, CE-159 y CE-161 son una gama de paquetes CMOS de RAM de entre 2 y 16 K, algunos de los cuales contienen ROM programable. El CE-153 Software Board es un teclado al tacto de 140 teclas para entrada formateada.

Chips de RAM
Proporcionan los 8,5 K de memoria, de los cuales hay 6,6 K disponibles para el usuario

Chips de visualización
Estos cuatro chips tratan la visualización en la pantalla LCD

Placa de circuito impreso
Para conseguir la máxima densidad, la placa se ha dividido en dos mitades, conectadas mediante un par de cables planos. Observe la escasez de componentes: éstos se han apiñado en los microchips CMOS

Chip de E/S
Maneja la gestión de los periféricos externos, tales como la impresora/cassette

SHARP PC-1500A

DIMENSIONES
195 × 85 × 25 mm

PESO
375 g

MEMORIA
8,5 K de RAM

PANTALLA
LCD de 1 × 26 caracteres

OBSERVACIONES
Calendario y reloj incorporados; buen BASIC; memoria ampliable; enorme gama de periféricos

SHARP PC-1251

DIMENSIONES
135 × 70 × 10 mm

PESO
115 g

MEMORIA
4 K de RAM

PANTALLA
LCD de 1 × 24 caracteres

OBSERVACIONES
Tres modalidades de funcionamiento; 18 teclas definibles por el usuario; dispone de cassette/impresora

Sharp PC-1251

Sharp considera esta calculadora como su sistema de gestión de uso diario (probablemente más para el ingeniero y el científico que para el administrador); las diminutas teclas de letras de su teclado QWERTY hacen que la entrada de textos resulte difícil y cansada

Sharp PC-1500A

Éste es un ordenador de bolsillo extraordinariamente flexible y del tamaño de una calculadora; sus argumentos de venta son su potencia y su equipo opcional, y podría en efecto contribuir en gran medida a aliviar la cuota de trabajo de oficina. Sin embargo, es probable que se lo considere sólo como un juguete para ejecutivos y una calculadora notable



Chris Stevens

Dibujo de imágenes

Esta vez diseñaremos visualizaciones de pantalla para el Spectrum con el fin de ilustrar dos escenarios de "Digitaya"

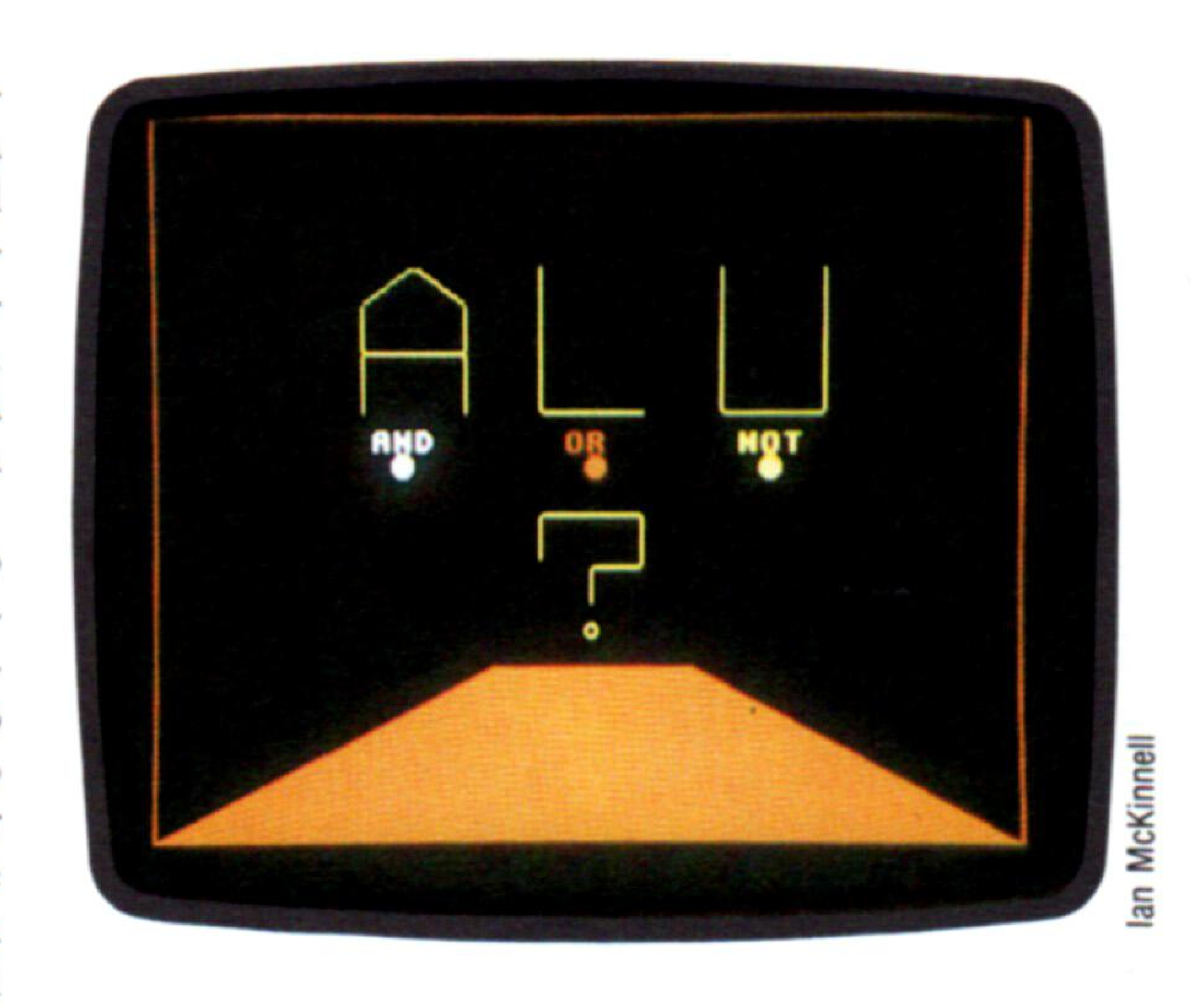
El diseño de la pantalla ALU implica el desplazamiento de las letras A, L y U hasta el centro de la pantalla utilizando gráficos en alta resolución. En el BBC Micro, este desplazamiento se efectuaba dibujando la letra desde un punto de partida especificado empleando instrucciones para dibujo relativo, borrándolas luego, desplazando el punto de partida y repitiendo todo el procedimiento. La misma idea se puede aplicar en la versión para el Spectrum.

La instrucción DRAW del Spectrum permite sólo el dibujo relativo (es decir, partiendo desde el último punto especificado), pero es ideal para esta particular aplicación de desplazamiento. Trazando (PLOT) un punto de partida inicial y llevando luego a cabo una serie de instrucciones DRAW para crear la forma de la letra, podemos desplazar fácilmente todo el diseño completo de la letra alrededor de la pantalla, con sólo cambiar las coordenadas del punto trazado inicialmente. El borrado se puede obtener dibujando la misma forma en la misma posición, pero con todos los colores invertidos. Este efecto se activa mediante INVERSE 1 y se desactiva mediante INVERSE 0. Por lo tanto, para cada posición que ocupe la letra la dibujaremos dos veces: una vez con INVERSE 0, para hacer aparecer la forma, y otra vez con INVERSE 1 para borrarla.

Si tomamos el ejemplo de la letra A, que se desplaza desde la izquierda, podemos colocar todas estas instrucciones dentro de un bucle FOR...NEXT. Este bucle incrementa el valor de la coordenada x del punto trazado inicialmente para la forma. Anidada en el interior de este bucle hay una segunda estructura FOR...NEXT que simplemente lleva a cabo dos veces las instrucciones de dibujo. El último valor de x es 55, que denota la posición de descanso final de la letra en la pantalla. Obviamente, no deseamos borrar la versión final de la letra, de modo que se inserta una condición para asegurar que la letra se borre (cambiando a INVERSE 1) sólo si la coordenada x es menor que 55. Los principios analizados aquí también se aplican a las otras dos letras para hacer que la L se desplace hacia arriba de la pantalla y la U lo haga desde la derecha.

Diseño esquemático de la ALU

Cuando se diseña una pantalla gráfica es importante hacer un esquema del diseño sobre papel y realizar un cálculo inicial de los valores de las coordenadas que tendrá cada forma en la pantalla. Además, también se han de situar todas las letras a imprimir en la pantalla, en términos de filas y columnas. La instantánea de la pantalla nos muestra tal diseño,



Ian McKinnell

con las dimensiones de ésta en unidades de gráficos y de caracteres.

Las palabras AND, OR y NOT se visualizan en la pantalla utilizando la instrucción PRINT AT r,c: siendo r el número de filas desde la parte superior de la pantalla, e indicando c el número de columnas desde el margen izquierdo. Los botones se dibujan empleando CIRCLE x,y,r, donde se especifican las coordenadas del centro y la longitud del radio.

Completadas las rutinas de dibujo, el programa espera a que se pulse una tecla antes de restablecer INK y PAPER a los colores originales y limpiar la pantalla. Después retorna a la rutina ALU principal. La pulsación de tecla se trata mediante INKEY$; de no pulsarse ninguna tecla, entonces se repite la condición. Para llamar a esta subrutina se debe insertar esta línea en el programa *Digitaya*:

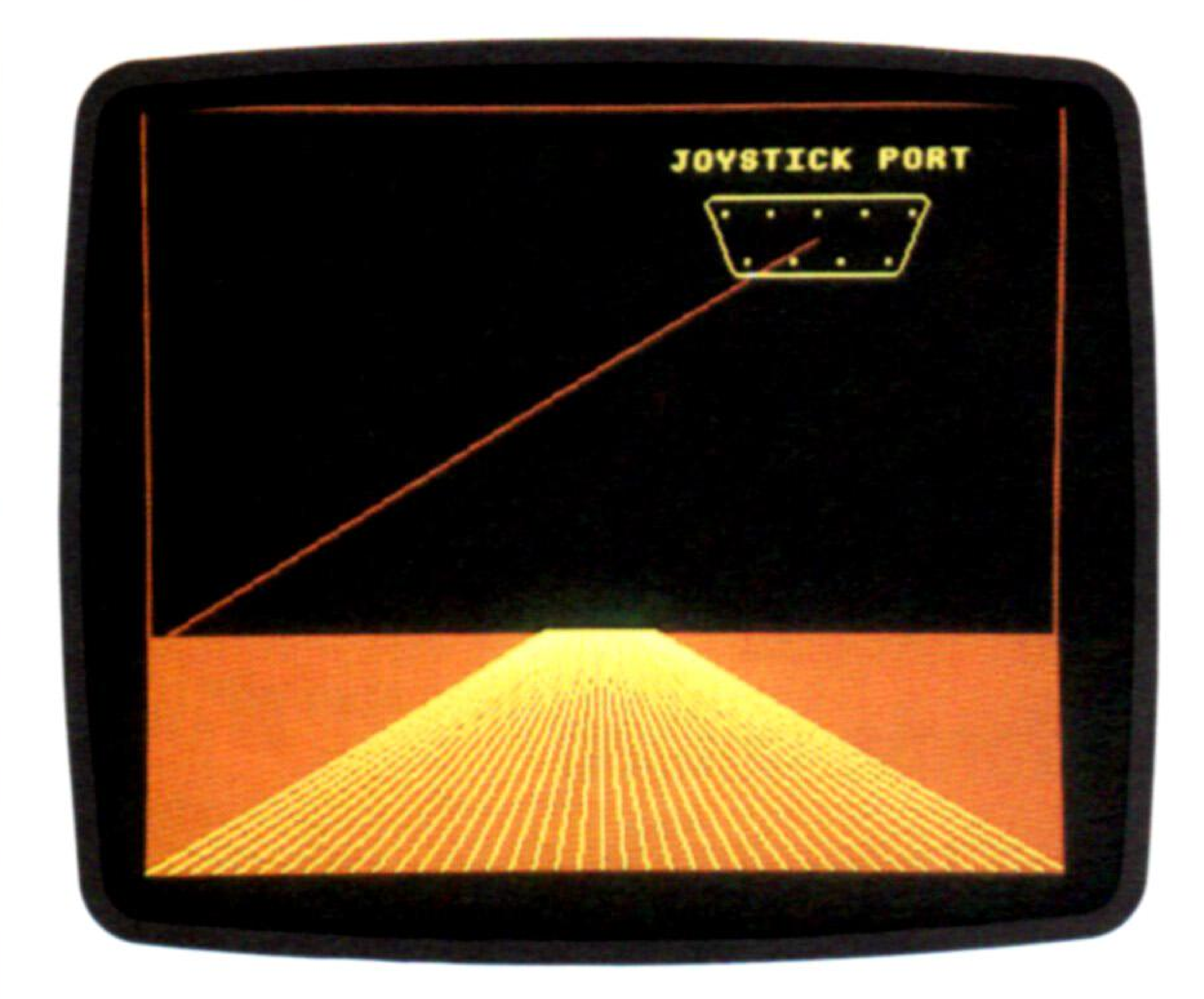
```
4565 GOSUB 7000:REM S/R IMAGEN ALU
```

La pantalla de la puerta para la palanca de mando está diseñada para disparar rayos láser desde el centro de un conector para palanca de mando. Las patillas del conector son caracteres de punto impresos en la pantalla y el contorno tipo D se dibuja empleando gráficos en alta resolución. Para conferirle a la imagen una sensación de profundidad, se dibuja en el primer plano una serie de líneas escalonadas. El punto de partida de cada línea se halla en la línea del horizonte y se selecciona mediante una instrucción PLOT. El final de cada línea está en la parte inferior de la pantalla. Las líneas están separadas por intervalos de una unidad en el horizonte, ampliándose la separación a siete unidades en la parte inferior de la pantalla.

El hecho de que la instrucción DRAW del Spectrum sea relativa hace que la rutina sea ligeramente más complicada que si pudiéramos especificar un

punto final absoluto. Si la coordenada x del punto de partida de la línea situada más a la izquierda es 111, entonces debemos efectuar un cálculo basado en ello para hallar el desplazamiento relativo hasta el punto final. El bucle FOR...NEXT de las líneas 8030-8060 muestra este cálculo.

Como antes, es útil esquematizar las dimensiones y las coordenadas del diseño sobre el papel antes de empezar a escribir el código. La instantánea de la pantalla nos muestra tal diseño:



Los rayos láser se dibujan desde la puerta para la palanca de mando mediante el desplazamiento hasta un punto del centro de la puerta y dibujando (DRAW) luego una línea hacia un punto del horizonte escogido al azar, utilizando un color INK seleccionado al azar. Repitiendo el procedimiento con INVERSE 1, podemos borrar la línea, hacer que el haz aparezca sólo durante un breve intervalo (creando un efecto de relámpago). Sin embargo, al borrar la línea surge un problema. Dado que el rayo se dibuja desde un punto situado en el centro de la puerta, atraviesa los gráficos dibujados previamente y que representan a la puerta propiamente dicha. Cuando se borra la línea aparecen agujeros en el gráfico de la puerta, y, por consiguiente, al borrar la línea es necesario volver a dibujarlo.

Aun cuando el final de cada línea dibujada desde la puerta para palanca de mando se detiene justo antes de la línea del horizonte, éste se ve afectado. Debido a la forma en que el Spectrum controla el color, la porción del horizonte más próxima al punto donde termina el rayo adquiere el mismo color que el utilizado para dibujar la línea. Ello se debe a que el Spectrum sólo puede soportar un color INK y un color PAPER dentro de una celda de carácter; cualquier gráfico que estuviera ya presente dentro de la celda tomará al color de primer plano del nuevo color INK empleado en la celda. Por consiguiente, además de volver a dibujar la puerta para palanca de mando, también se debe volver a dibujar la línea del horizonte después de borrar un rayo. La rutina continúa disparando rayos láser hasta que se efectúa una pulsación de tecla, en cuyo momento el control de programa retorna a la rutina de la puerta para palanca de mando principal, después de haber restablecido los colores INK y PAPER normales. Para llamar a esta subrutina se debe insertar esta línea:

```
3845 GOSUB 8000:REM IMAGEN PUERTA PALANCA
     MANDO
```

Pantalla ALU

```
7000 REM *** s/r imagen alu ****
7010 INK 6:PAPER 0:CLS
7015 :
7017 REM **** letra A ****
7020 FOR x=0 TO 55 STEP 5
7030 INVERSE 0
7040 FOR i=1 TO 2
7050 PLOT x,100
7060 DRAW 0,30
7070 DRAW 15,20
7080 DRAW 15,-20
7090 DRAW 0,-30
7095 DRAW 0,20
7096 DRAW -30,0
7110 IF x>55 THEN INVERSE 1
7115 NEXT i
7120 NEXT x
7130 :
7140 REM **** letra L ****
7150 FOR y=100 TO 150 STEP 5
7152 INVERSE 0
7155 FOR i=1 TO 2
7160 PLOT 113,y
7170 DRAW 0,-50
7180 DRAW 30,0
7190 IF y<150 THEN INVERSE 1
7200 NEXT i
7210 NEXT y
7220 :
7230 REM **** letra U ****
7240 FOR x=225 TO 170 STEP -5
7250 INVERSE 0
7260 FOR i=1 TO 2
7270 PLOT x,150
7280 DRAW 0,-50
7290 DRAW 30,0
7300 DRAW 0,50
7310 IF x>170 INVERSE 1
7320 NEXT i
7330 NEXT x
7340 :
7350 REM **** botones ****
7360 PRINT AT 10,7;"AND"
7370 PRINT AT 10,15;"OR"
7380 PRINT AT 10,22;"NOT"
7390 INK 3: CIRCLE 70,80,5
7400 INK 4:CIRCLE 128,80,5
7410 INK 5: CIRCLE 185,80,5
7420 :
7430 REM **** signo ? ****
7435 INK 6
7440 PLOT 113,45
7450 DRAW 0,15
7460 DRAW 30,0
7470 DRAW 0,-20
7480 DRAW -15,0
7490 DRAW 0,-7
7500 FOR r=6 TO 0 STEP -2
7510 CIRCLE 128,23,r
7520 NEXT r
7530 :
7540 IF INKEY$="" THEN GO TO 7540
7550 INK 0: PAPER 7: CLS
7560 RETURN
```

Pant. puerta para pal. mando

```
8000 REM **** s/r imagen puerta pal. mando ****
8010 INK 6: PAPER 0: CLS
8020 REM **** primer plano ****
8030 FOR n=1 TO 31
8040 PLOT 112+n,50
8050 DRAW 7*n-112,-50
8060 NEXT n
8070 :
8080 REM **** horizonte ****
8085 INK 6: INVERSE 0
8090 PLOT 0,50
8100 DRAW 255,0
8110 :
8120 REM **** puerta ****
8130 PRINT AT 1,18;"JOYSTICK PORT"
8140 PRINT AT 3,20;". . . . ."
8150 PRINT AT 5,21;". . . ."
8160 PLO 158,152
8170 DRAW 75,0
8180 DRAW 1,-1
8190 DRAW 1,-1
8200 DRAW 0,-1
8210 DRAW -1,-1
8220 DRAW -10,-25
8230 DRAW -2,-2
8240 DRAW -52,0
8250 DRAW -2,2
8260 DRAW -10,25
8270 DRAW -1,1
8280 DRAW -1,1
8290 DRAW 0,1
8300 DRAW 1,1
8310 :
8320 REM **** disparo ****
8340 INK RND*7
8350 LET x=RND*255-194
8360 LET y=-86
8365 INVERSE 0
8367 FOR i=1 TO 2
8370 PLOT 194,136
8380 DRAW x,y
8385 INVERSE 1
8387 NEXT i
8390 :
8400 REM **** comprobar tecla ****
8410 IF INKEY$="" THEN GO TO 8080
8415 INVERSE 0
8420 INK 0: PAPER 7: CLS
8430 RETURN
```

Patrones de enrejado

Veamos cómo se pueden crear las cuadrículas en que se basan los patrones en dos dimensiones

Si, en vez de trasladar el motivo que definimos en el capítulo anterior a lo largo de una línea, permitimos que se produzcan simultáneamente dos traslaciones no paralelas, entonces nuestro patrón se convierte en bidimensional. Comenzaremos a investigar este conjunto de patrones utilizando como unidad un único punto.

```
TO PUNTO
  PD FD 1 BK 1 PU
END
```

El procedimiento para llevar a cabo estas traslaciones simultáneas se define del siguiente modo:

```
TO CUADRICULA :PARTIDAX :PARTIDAY :XPASO
  :YPASO :ANGULO DRAW HT PU
  SETXY :PARTIDAX :PARTIDAY SETH 0
  REPEAT 3 [LINEA :XPASO ABAJO :YPASO
  :ANGULO]
END
```

Este procedimiento dibuja una cuadrícula de nueve puntos. Las entradas dan las coordenadas del punto de partida, el tamaño de los pasos X y Y, y la orientación de los puntos correspondientes de la siguiente fila desde la fila actual.

Éste es el procedimiento LINEA:

```
TO LINEA :X
  REPEAT 3 [UNIDAD SETX XCOR+:X]
  SETX XCOR−3*:X
END
```

dibuja una única línea de tres unidades a través de la pantalla, y después devuelve la tortuga a su punto de partida.

De momento, el procedimiento UNIDAD es simplemente un punto:

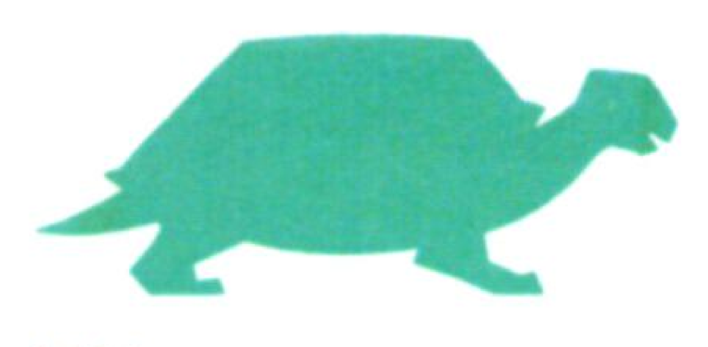
```
TO UNIDAD
  PUNTO
END
```

Otro procedimiento:

```
TO ABAJO :Y :A
  SETH :A
  FD :Y
  SETH 0
END
```

desplaza la tortuga hasta la fila siguiente (tal como lo hemos utilizado, ha supuesto el desplazamiento hacia "abajo" hasta la siguiente fila, de ahí el nombre del procedimiento), y luego restaura la orientación. En el diagrama se ofrecen los cinco tipos de enrejado plano, junto con los procedimientos en LOGO para dibujarlos.

A partir de combinaciones de estas cuadrículas básicas se pueden obtener muchos patrones diferentes, si bien probablemente sea más interesante modificar el procedimiento para que dibuje una línea en diversos ángulos en vez de recta a través de la pantalla.



Otra línea de investigación es ver cómo se puede perfeccionar la simetría de cada una de las cuadrículas añadiendo a la unidad dibujada en cada punto diversas formas de simetría. Existen 17 de tales patrones y todos ellos se pueden apreciar en el segundo diagrama. Un método obvio para dibujar estas 17 posibilidades consiste en reemplazar la instrucción UNIDAD del procedimiento LINEA por un procedimiento que dibuje la forma en ese punto.

La forma unidad

Las formas unidad se consiguen a partir de un motivo básico junto con diversas reflexiones y rotaciones. A modo de demostración, vamos a definir nuestro motivo básico, al cual llamaremos LIT. (Como antes, éste es de estado transparente y no emplea ningún procedimiento.)

```
TO LIT
  PD
  FD 15
  RT 90
  FD 5
  BK 5
  LT 90
  BK 15
  PU
END
```

Podemos utilizar los procedimientos que desarrollamos en el capítulo anterior para crear dos procedimientos: MOTIVO y su imagen de espejo, I.MOTIVO:

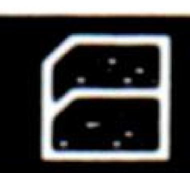
```
DEFINE "MOTIVO TEXTO "LIT
DEFINE "I. MOTIVO REESCRIBIR "LIT
```

Ahora podemos definir las formas unidad. Por ejemplo, las unidades para los patrones 7 y 17 serían las siguientes:

```
TO UNIDAD7
  LT 90 MOTIVO RT 90
END

TO UNIDAD17
  RT 30
  REPEAT 6 [MOTIVO I.MOTIVO RT 60]
  LT 30
END
```

Si vuelve a observar el procedimiento LINEA, verá que el mismo ejecuta el procedimiento UNIDAD. Por consiguiente, con el fin de dibujar el patrón 7, por ejemplo, se debe modificar el procedimiento UNIDAD de modo que rece UNIDAD7. Esto lo hacemos mediante el empleo de DEFINIR.UNIDAD7, donde:



```
TO DEFINIR.UNIDAD :NUM
  DEFINE "UNIDAD TEXT PALABRA "UNIDAD :NUM
END
```

Ahora ejecutaremos al mismo tiempo las partes de un patrón para dibujar la cuadrícula y dibujar la unidad.

Un procedimiento denominado PAT nos permite hacerlo:

```
TO PAT :CUADRICULA :NUM :PROC
  DEFINE "MOTIVO TEXT :PROC
  DEFINE "I.MOTIVO REESCRIBIR :PROC
  DEFINIR.UNIDAD :NUM
  RUN (LIST :CUADRICULA)
  ERASE MOTIVO
  ERASE I. MOTIVO
  ERASE UNIDAD
END
```

Para dibujar el patrón 17 ahora digitaremos:

```
PAT "HEX 17 "LIT
```

Esto dibuja una cuadrícula hexagonal, con UNIDAD17 en cada punto, utilizando LIT como motivo básico.

Este método funciona bien para todos los patrones, excepto para el 4, el 6, el 7 y el 12. En estos casos, la forma unidad no es la misma en cada punto, sino que sufre una transformación (reflexión, rotación o ambas a la vez). Una forma de tratar esta cuestión consiste en incorporar estas transformaciones a los procedimientos LINEA y ABAJO. De modo que vamos a definir TRANSX como la transformación a aplicar a la traslación básica a través de la pantalla, y TRANSY será la transformación a aplicar entre filas. Entonces LINEA y ABAJO se convierten en:

```
TO LINEA :X
  REPEAT 3 [UNIDAD SETX XCOR XCOR+:X
  TRANSX]
  SETX XCOR-3*:X
END

TO ABAJO :Y :A
  SETH :A
  FD :Y
  SETH 0
  TRANSY
END
```

Ahora definimos el patrón 7 como:

```
TO PATRON7 :PROC
  DEFINE "TRANSX[[][REFLEXION RT 180]]
  DEFINE "TRANSY[[][]]
  PAT "RECT 7 :PROC
  ERASE TRANSX
  ERASE TRANSY
END
```

Para utilizarlo, entre PATRON7 "PATA. Tras ejecutar el procedimiento anterior, TRANSX se habrá definido como:

```
TO TRANSX
  REFLEXION
  RT 180
END
```

REFLEXION se utiliza para reflejar el patrón unidad. Este procedimiento se define reescribiendo el procedimiento UNIDAD:

Los diecisiete grupos planos

Clave:
P = enrejado de paralelogramo
R = enrejado rectangular
C = enrejado romboidal
S = enrejado cuadrangular
H = enrejado hexagonal
ROS = orden rotacional de simetría
M = reflexión de espejo
G = reflexión con deslizamiento

Paralelogramo

P1(ROS=1)

P2(ROS=2)

Rectangular

RM(ROS=1)

RG(ROS=1)

RMM(ROS=2)

RMG(ROS=2)

RGG(ROS=2)

Romboidal

CM(ROS=1)

CMM(ROS=2)

Cuadrangular

S4(ROS=4)

S4M(ROS=4)

S4G(ROS=4)

Hexagonal

H3(ROS=3)

H3M1(ROS=3)

H3M2(ROS=3)

H6(ROS=6)

H6M(ROS=6)

Inicio del enrejado
Los 17 patrones planos que vemos aquí están agrupados de acuerdo a sus patrones de enrejado básicos. H3M1 y H3M2, por ejemplo, se basan en un enrejado hexagonal, poseen un orden de simetría de 3 e incorporan una reflexión de espejo. Sólo difieren en que el primero tiene un eje de reflexión a lo largo de las líneas de la cuadrícula, y el segundo hacia abajo del eje vertical.

Ian McKinnell

Los cinco tipos de enrejado de plano

Paralelogramo

```
TO PARALEL
  CUADR (-60) 90 80 50 205
END
```

Romboidal

```
TO ROMB
  CUADR (-30) 90 80 80 225
END
```

Rectangular

```
TO RECT
  CUADR (-80) 90 80 50 180
END
```

Cuadrangular

```
TO CUADRADO
  CUADR (-80) 90 80 80 180
END
```

Hexagonal

```
TO HEX
  CUADR (-30) 90 80 80 210
END
```

```
TO REFLEXION
  DEFINE "UNIDAD REESCRIBIR "UNIDAD
END
```

La reescritura supone ahora sustituir RT por LT, y viceversa, así como MOTIVO por I.MOTIVO y viceversa.

Nuestra versión previa del procedimiento de reescritura sólo intercambiaba RT y LT. Para modificar esto, todo cuanto hemos de hacer es modificar CAMBIAR.PALABRA, que ahora se convierte en:

```
TO CAMBIAR.PALABRA :PALABRA
  IF (ANYOF :PALABRA="RT :PALABRA="RIGHT)
  THEN OUTPUT "LEFT
  IF(ANYOF :PALABRA="LT :PALABRA="LEFT)
  THEN OUTPUT "RIGHT
  IF :PALABRA="MOTIVO THEN OUTPUT
  "I.MOTIVO
  IF :PALABRA="I.MOTIVO THEN OUTPUT
  "MOTIVO OUTPUT :PALABRA
END
```

Otra forma de abordar este problema sería utilizar la versión de REESCRIBIR que también modifica los subprocedimientos del procedimiento de entrada. Esta versión la ofrecemos entre las respuestas.

Para la mayoría de los patrones el movimiento entre puntos es simplemente una traslación, y no hay que llevar a cabo ninguna otra transformación. TRANSX y TRANSY, en consecuencia, no hacen nada. PATRON17 constituye un ejemplo de este tipo:

```
TO PATRON17 :PROC
  DEFINE "TRANSX[[][]]
  DEFINE "TRANSY[[][]]
  PAT "HEX 17 :PROC
  ERASE TRANSX
  ERASE TRANSY
END
```

Habiendo cubierto todas las posibilidades básicas, dejamos en sus manos la definición del resto de los patrones.

Complementos al LOGO

Para todas las versiones LCSI:

Emplee CS en lugar de DRAW
Emplee OR en lugar de ANYOF
SETPOS, seguido de una lista, se utiliza en lugar de SETXY
IF posee una sintaxis diferente:

```
IF :PALABRA=MOTIVO[OUTPUT "I.MOTIVO]
```

En el LOGO Atari TEXT y DEFINE no existen como primitivas, si bien en el manual se ofrece un método para definirlas

Respuestas a los ejercicios

1. Para hacer girar una forma alrededor del punto (X,Y) en un ángulo de A grados:

```
TO ROTAR :X :Y :A
  PU
  MAKE "O ORIENTACION
  MAKE "XANT XCOR
  MAKE "YANT YCOR
  MAKE "R SQRT (:XANT-:X)*(:XANT-
  :X)+(:YANT-:Y)*(:YANT-:Y)
  PU
  SETXY :X :Y
  SETH TOWARDS :XANT :YANT
  RT :A
  FD :R
  SETH :0+:A
  PD
END
```

2. Un procedimiento para reescribir que también reescriba subprocedimientos.

```
MAKE "PROCEDS.HECHOS[]

TO REESCRIBIR :PROC
  MAKE "PROCEDS.HECHOS FPUT :PROC
  :PROCEDS.HECHOS
  OUTPUT REESCRIBIR.PROC TEXT :PROC
END

TO REESCRIBIR.PROC :TEXTO
  IF :TEXTO=[]THEN OUTPUT[]
  OUTPUT FPUT REESCRIBIR.LINEA FIRST
  :TEXTO REESCRIBIR.PROC BUTFIRST
  :TEXTO
END

TO REESCRIBIR.LINEA :LINEA
  IF :LINEA=[] THEN OUTPUT[]
  IF LIST? FIRST :LINEA THEN OUTPUT FPUT
  REESCRIBIR.LINEA FIRST :LINEA
  REESCRIBIR. LINEA BUTFIRST :LINEA
  OUTPUT FPUT CAMBIAR.PALABRA FIRST
  :LINEA REESCRIBIR.LINEA BUTFIRST :LINEA
END

TO CAMBIAR.PALABRA :PALABRA
  IF (ANYOF :PALABRA="RT :PALABRA=
  "RIGHT)THEN OUTPUT "LEFT
  IF (ANYOF :PALABRA="LT :PALABRA=
  "LEFT) THEN OUTPUT "RIGHT
  IF PROCEDIMIENTO? :PALABRA THEN
  SUBPROCEDIMIENTO :PALABRA OUTPUT
  WORD "£:PALABRA
  OUTPUT :PALABRA
END

TO PROCEDIMIENTO?:NOMBRE
  IF NUMBER? :NOMBRE OUTPUT "FALSE
  IF LIST? :NOMBRE OUTPUT "FALSE
  TEST WORD? :NOMBRE
  IFTRUE IF WORD? TEXT :NOMBRE OUTPUT
  "FALSE ELSE IF NOT
  (TEXT :NOMBRE=[])
  OUTPUT "TRUE
  OUTPUT "FALSE
END

TO SUBPROCEDIMIENTO :PALABRA
  IF MEMBER? :PALABRA :PRODECS.HECHOS
  THEN STOP
  DEFINE (WORD "£:PALABRA)REESCRIBIR
  :PALABRA
END
```



Adagio ma non troppo

Vamos a crear una rutina que desplace horizontalmente un dibujo de fondo en una pantalla del Commodore 64. Es obvia su utilidad en programas de juegos

El VIC o controlador de video del Commodore puede desplazar hasta ocho pixels la visualización en pantalla, en las dos direcciones. El desplazamiento horizontal se regula por medio de los tres bits inferiores correspondientes al registro del VIC situado en la posición 53270 ($D016). Si se van asignando valores progresivamente a estos tres bits del 7 al 0, la pantalla se va desplazando un pixel a la izquierda a cada valor. En BASIC se emplearía esta sentencia:

```
POKE 53270,(PEEK(53270)AND 248)+P
```

donde P tiene un valor entre 0 y 7.

Combinando esta facilidad con una rutina en código máquina para desplazar todos los datos de la pantalla una posición a la izquierda e introducir una nueva columna de datos en el margen derecho, es posible obtener un suave efecto de desplazamiento. La pantalla ha de reducirse a 38 columnas (en lugar de las normales 40 columnas) para que los datos aparezcan y desaparezcan de nuestra vista en un pausado desfile. El cambio al modo 38 columnas se realiza poniendo a cero el bit 3 del registro de los desplazamientos horizontales. En BASIC se hace así:

```
POKE 53270,PEEK(53270)AND247
```

La pantalla volverá a sus 40 columnas normales poniendo de nuevo el bit 3 a uno.

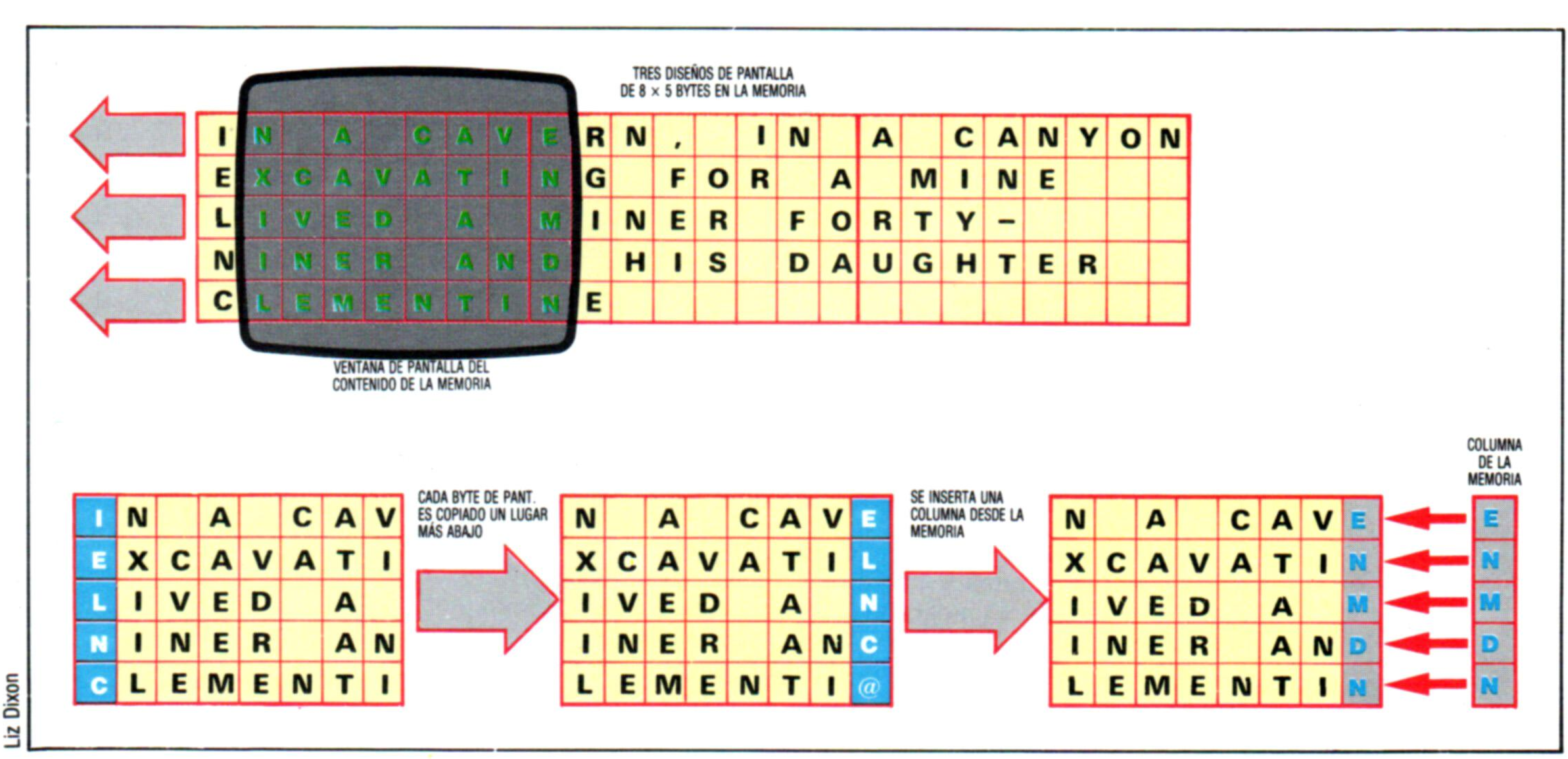
El diagrama de flujo muestra las diferentes tareas a realizar para producir ese suave movimiento horizontal. Importa añadir que si movemos o insertamos datos de pantalla, hay que hacer similares cambios en los datos del color.

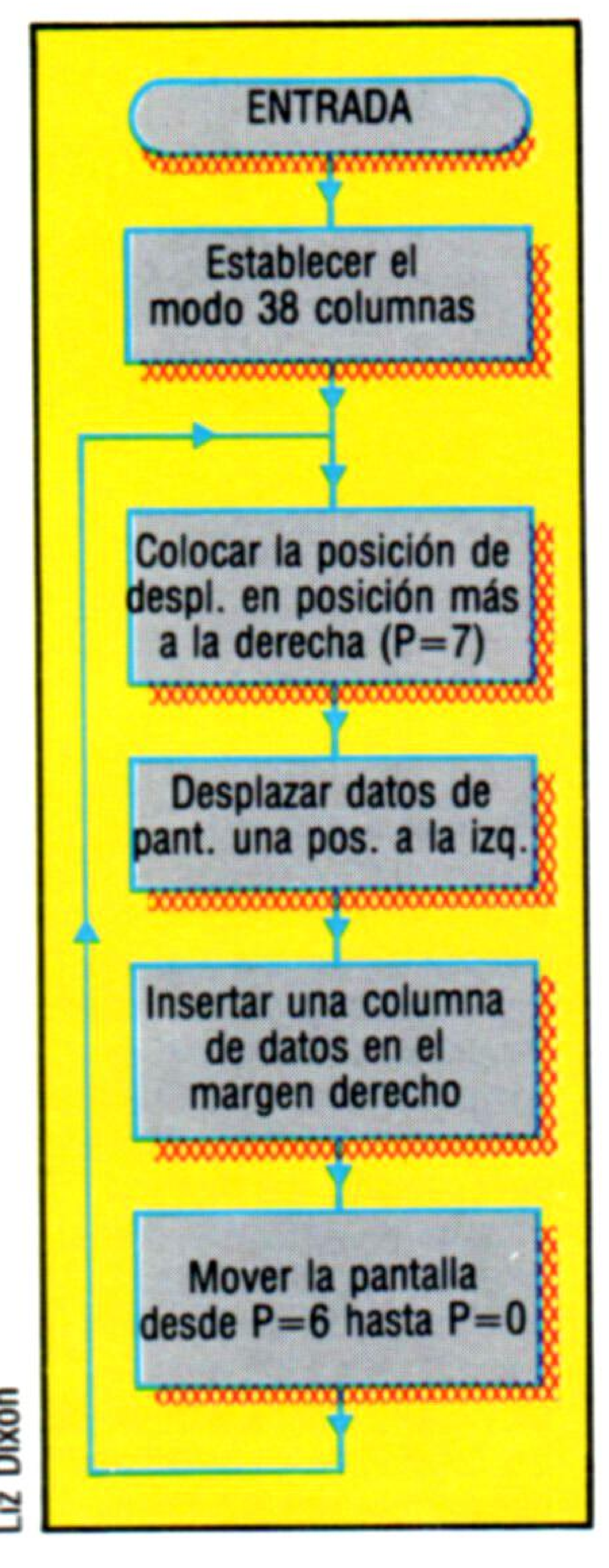
Cambio de los datos de pantalla

Se trata de una tarea en principio fácil. Los datos de pantalla se suelen guardar a partir de la posición 1024 ($0400): los primeros 40 bytes constituyen la fila superior, los siguientes 40 bytes la fila inmediatamente inferior, y así sucesivamente. Para dar la sensación de que los datos se mueven un lugar a la izquierda, basta con llevar cada byte de dichos datos al byte que se encuentra debajo de la posición original. Este fragmento de la rutina utiliza punteros de página cero y el direccionamiento indirecto para colocar cada byte de pantalla y color un byte más abajo en la memoria.

Llamando SB a la dirección *B*ase del área de la pantalla (*S*creen), la última posición de la fila superior será SB+39, la última de la fila inmediata inferior será SB+79, etc. Para que los datos a desplazar sobre la pantalla puedan almacenarse de modo similar en la memoria (o sea, en paquetes de 1 000 bytes), los datos a insertar por el lado derecho de la pantalla serán los bytes primero, 41-ésimo, 81-ésimo, etc., del área que reservamos para tales datos. Esto lo ilustramos con el siguiente dibujo:

Muévase a voluntad
El desplazamiento de las visualizaciones sobre la pantalla y desde la memoria pasa por tres fases principales. Primero, cada byte de la memoria de pantalla se mueve hacia abajo una posición. Debido al diseño de la pantalla, sus filas son tratadas en la memoria como bytes consecutivos. Esto produce el efecto en pantalla como si cada carácter se moviera un lugar a la izquierda, a excepción de los caracteres que aparecen en la columna extrema izquierda de la pantalla. En esta columna cada carácter parece como si apareciera en la misma posición pero en la columna extrema derecha. El carácter superior de la columna extrema izquierda desaparece durante el proceso dando entrada a un carácter extraño por la esquina inferior izquierda. La segunda fase se encarga de copiar la columna que interesa tomándola de la memoria y llevándola a la columna extrema derecha de la pantalla. Los registros del chip VIC a cargo del desplazamiento pueden accionarse para dar la sensación de que la pantalla se desplaza un pixel cada vez dentro del área visible





Liz Dixon

Poco a poco, suavemente
Para conseguir un movimiento suave podemos hacer uso de una facilidad particular del VIC. Este chip está dotado de unos registros especiales de desplazamiento que permiten a la pantalla visible moverse de su posición inicial respecto al encuadre. Es posible producir pixels individuales tanto en dirección horizontal como vertical. Combinando este efecto con el copiado de caracteres en código máquina, podemos obtener un suave movimiento sobre una pantalla reducida de 38 columnas

Al comienzo se establecerá un puntero que apunte al byte que está al comienzo del área de memoria que deseamos desplazar en pantalla. Una vez obtenido el desplazamiento, el puntero se incrementará en una unidad para que copie la columna siguiente en el margen derecho de la pantalla, desde donde puede desplazarse a la izquierda. Tras reiterar el proceso 40 veces consecutivas, se habrá obtenido un desplazamiento de todos los datos de la pantalla. El puntero será entonces incrementado en 960 unidades (1000-40) para apuntar al inicio de la pantalla siguiente.

Este proceso deberá ser repetido para el área de los datos de color. En resumen, haremos que la dirección de cada byte en el mapa del color tenga un desplazamiento hacia la dirección del byte correspondiente en el mapa de datos de pantalla. El proceso será repetido para tantas pantallas de datos como se hayan diseñado y guardado consecutivamente en memoria. Para utilizar la rutina del desplazamiento, hay que enviar algunas informaciones previas. La rutina necesita saber:

1) La dir. de inicio del área de memoria donde se guardan los datos de la pant. a desplazar.
2) El desplazamiento a los correspondientes datos de color.
3) El número de pantallas de datos a desplazar.
4) Un valor de retardo que sirva para ralentizar la operación del movimiento.

Estos datos han de ser colocados (POKE) en posiciones reservadas dentro del prog. en cód. máquina.

Prog. de llamada en BASIC

```
10 REM ********************************
20 REM ********************************
30 REM **                            **
40 REM **  PROGRAMA BASIC DE LLAMADA  **
50 REM **   A RUTINA DESPLAZAMIENTO   **
60 REM **                            **
70 REM ********************************
80 REM ********************************
90 :
95 DN=8 : REM PARA CASS DN=1
100 IF A=0 THEN A=1 : LOAD "SCROLL.HEX",DN,1
110 POKE 55,0 : POKE 56,32 : CLR : REM EXTREMO INFERIOR MEMORIA
115 REM GOSUB 1000 : REM ESTABLECE VISUALIZACION SIMPLE
120 :
130 LMEM    =49664:     REM INICIO MEMORIA
140 HMEM    =49665:     REM AREA
150 LCOFF   =49666:     REM DESPLAZ. AL COLOR
160 HCOFF   =49667:     REM MAPA
170 NMSCR   =49668:     REM NUMERO PANTALLAS (SCREENS)
190 DELAY   =49669:     REM VALOR DEL RETARDO (DELAY)
200 SCROLL  =49670:     REM DIRECCION INICIO PROGRAMA
210 :
220 REM PRINT CHR$(147) : REM BORRA PANTALLA
230 INPUT "DIRECCION INICIO EN DECIMAL";SA
240 HS=INT(SA/256):LS=SA-HS*256
250 POKE LMEM,LS: POKE HMEM,HS
260 :
270 INPUT "NUMERO PANTALLAS";NS
290 POKE NMSCR,NS
300 :
310 INPUT "DESPLAZ EN DECIMAL AL MAPA COLOR" ;OS
320 HO=INT(OS/256):LO=OS-HO*256
330 POKE LCOFF,LO:POKE HCOFF,HO
340 :
350 INPUT "VALOR DEL RETARDO>256";DV
360 IF DV>255 OR DV<0 THEN 350
370 POKE DELAY,DV
380 :
390 SYS SCROLL
400 POKE 53270,PEEK (53270) OR B
```

El programa carga el código máquina en la memoria y pide la información necesaria a través de instrucciones INPUT. El programa secciona la información en la forma LO-HI donde sea necesario y la coloca (POKE) en los espacios de almacenamiento dispuestos al comienzo del programa. Posteriormente se llama a la rutina en código máquina.

Toda dirección de inicio, desplazamiento y número de pantallas deben ser especificadas, aunque los resultados no serán muy espectaculares si no se ha puesto ningún dibujo en el área de memoria especificada. Puede comprobarse el programa cargando y ejecutando el corto programa en BASIC que establece dos pantallas sencillas de datos que comienzan en la posición 8192. El desplazamiento al área de datos de color es 3 000 bytes. Para mover esta área de datos en la pantalla, hay que proporcionar la siguiente información en respuesta a las preguntas del programa que llama:

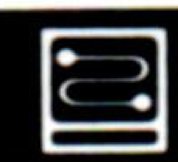
1) Dirección de inicio, en decimal:	8192
2) Desplazamiento del color:	3000
3) Número de pantallas:	2
4) Retardo:	255

Cargador en BASIC

```
10 REM ***************************
15 REM **    CARGADOR EN BASIC    **
20 REM **  PARA RUTINA DESPLAZ    **
30 REM **       HORIZONTAL        **
40 REM ***************************
50 :
60 FOR I=49670 TO 49945
70 READ A : POKE I,A
80 CC=CC+A
90 NEXT
92 READ CS:IF CS<>CC THEN PRINT "ERROR SUMA CONTROL":STOP
100 DATA173,22,208,41,247,141,22,208
110 DATA174,4,194,160,40,138,72,152,72
120 DATA173,22,208,41,248,24,105,7,141
130 DATA22,208,169,0,133,251,169,4,133
140 DATA252,169,0,133,253,169,216,133
150 DATA254,162,3,160,1,177,251,136
160 DATA145,251,200,177,253,136,145
170 DATA253,200,200,208,241,230,252
180 DATA230,254,177,251,198,252,136
190 DATA145,251,200,177.253.198,254
200 DATA136,145,253,230,252,230,254
210 DATA202,208,213,160,1,177,251,136
220 DATA145,251,200,177,253,136,145
230 DATA253,200,200,192,232,144,239
240 DATA173,0,194,133,253,173,1,194
250 DATA133,254,169,39,133,251,169,4
260 DATA133,252,32,241,194,173,0,194
270 DATA24,109,2,194,133,253,173,1,194
280 DATA109,3,194,133,254,169,39,133
290 DATA251,169,216,133,252,32,241,194
300 DATA162,6,173,22,208,41,248,141,22
310 DATA208,138,24,109,22,208,141,22
320 DATA208,172,5,194,136,208,253,202
330 DATA16,231,173,0,194,24,105,1,141
340 DATA0,194,173,1,194,105,0,141,1
350 DATA194,104,168,104,170,136,240,3
360 DATA76,19,194,173,0,194,24,105,192
370 DATA141,0,194,173,1,194,105,3,141
380 DATA1,194,202,240,3,76,17,194,96
390 DATA162,25,160,0,177,253,145,251
400 DATA202,240,29,165,251,24,105,40
410 DATA133,251,165,252,105,0,133,252
420 DATA165,253,24,105,40,133,253,165
430 DATA254,105,0,133,254,76,245,194
440 DATA96
450 DATA40227:REM*SUMA CONTROL*
```

Rutina para establecer la visualización

```
1000 REM **** ESTABLECE VISUALIZACION ****
1010 CL=3000:REM DESPLAZ AL MAPA COLOR
1020 SS=8192:REM INICIO MAPA VISUALIZ.
1030 FORI=SS TO SS+479
1040 POKE I,1:REM CODIGO PANT. DE LA "A"
1050 POKE+CL,1:REM BLANCO
1060 POKEI+480,2:REM CODIGO PANT. DE LA "B"
1070 POKEI+CL+480,14:REM AZUL CLARO
1080 NEXT
1085 FORI=SS+960TOSS+999
1090 POKEI,3:REM CODIGO PANT. DE LA "C"
1100 POKEI+CL,3:REM AZUL MARINO
1110 NEXT
1999 :
2020 SS=9192:REM SIGUIENTE INICIO PANT.
2030 FORI=SS TO SS+479
2040 POKEI,3:REM CODIGO PANT. DE LA "C"
2050 POKEI+CL,5:REM VERDE
2060 POKEI+480,4:REM CODIGO PANT. DE LA "D"
2070 POKEI+CL+480,0:REM NEGRO
2080 NEXT
2085 FORI=SS+960TOSS+999
2090 POKEI,5:REM CODIGO PANT. DE LA "E"
2100 POKEI+CL,2:REM ROJO
2110 NEXT
```

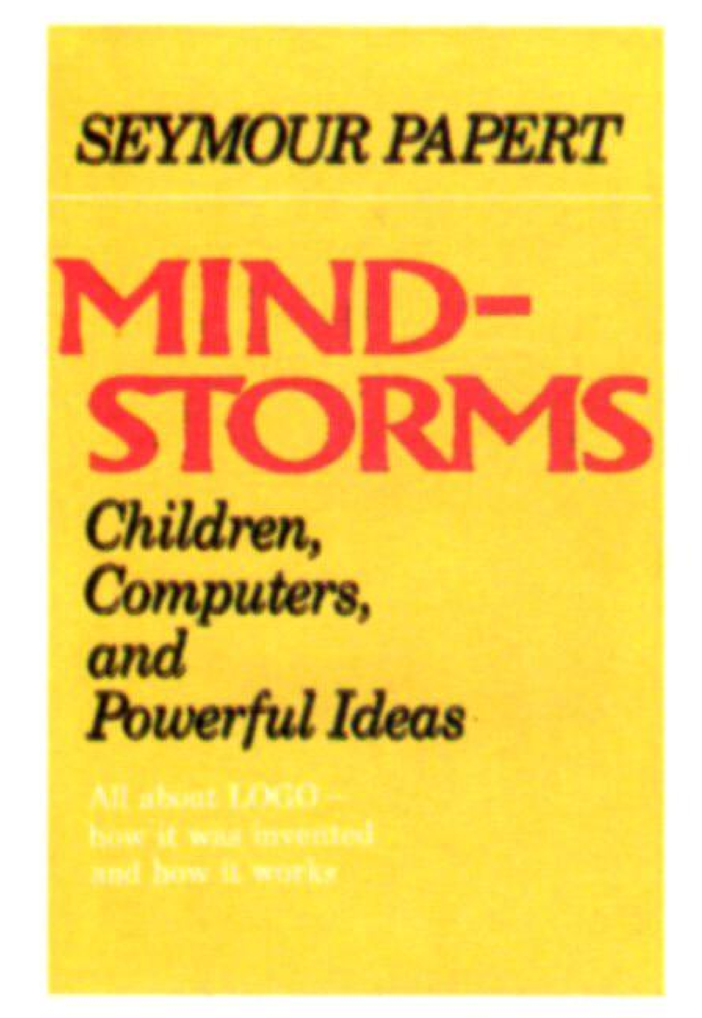


Movimiento horizontal

```
; ++++++++++++++++
; ++++++++++++++++
;++                ++
;++  MOV. HORIZONTAL  ++
;++                ++
;++ PARA COMMODORE 64 ++
;++                ++
; ++++++++++++++++
; ++++++++++++++++
;
SCRPTR=$FB              ;PAGINA 0
COLPTR=$FD              ;PUNTEROS COPIA
;
MEMPTR=$FD              ;PUNT P CERO DE MEMORIA
SCRLRG=$D016            ;REGISTRO DESPLAZ HORIZ
SCRNLO=$00              ;BYTE LO DE INICIO PANT
SCRNHI=$04              ;BYTE HI DE INICIO PANT
COLRLO=$00              ;BYTE LO DE INICIO COLOR
COLRHI=$D8              ;BYTE HI DE INICIO COLOR
BLOCKS=$03              ;BLOQUES DE BYTES 3*256
EXTRA=$E8               ;BYTES EXTRA HASTA 1000
NMCOLS=$28              ;NUM DE COLUMNAS
NMROWS=$19              ;NUM DE FILAS
;
*=$C200
;
MEMLO   *=*+1           ;INICIO MEMORIA QUE
MEMHI   *=*+1           ;SE HA DE DESPLAZAR
COFFLO  *=*+1           ;DESPLAZ AL MAPA DE COLOR
COFFHI  *=*+1
NMSCRN  *=*+1           ;NUM PANTALLAS
DELAY   *=*+1           ;VALOR BUCLE DEL RETARDO
;
;++++ ESTABLECE MODO 38 COLUMNAS ++++
;
        LDA SCRLRG
        AND #$F7
        STA SCRLRG
;
;++++ ESTABLECE PRIMERA POSICION A DESPLAZ ++++
;
        LDX NMSCRN
NEXSCR
        LDY #NMCOLS
START
        TXA
        PHA             ;COLOCA LOS REG X,Y
        TYA             ;EN LA PILA (PUSH)
        PHA
;
        LDA SCRLRG
        AND #$F8
        CLC
        ADC #$07
        STA SCRLRG
;
;++++ COPIA PANT/COLOR UN LUGAR IZQ ++++
;
        LDA #SCRNLO
        STA SCRPTR      ;ESTABLECE PAGINA 0
        LDA #SCRNHI
        STA SCRPTR+1    ;PUNTEROS PARA LA
        LDA #COLRLO
        STA COLPTR      ;COPIA
        LDA #COLRHI
        STA COLPTR+1
;
        LDX #BLOCKS
AGAIN
        LDY #$01
NEXT
        LDA (SCRPTR),Y
        DEY
        STA (SCRPTR),Y
        INY
        LDA (COLPTR),Y
        DEY
        STA (COLPTR),Y
        INY
        INY
        BNE NEXT
;
;++ TRASCRIBE BORDES PAGINA ++
        INC SCRPTR+1    ;INCR BYTES HI DE
        INC COLPTR+1    ;PUNTEROS PAGINA 0
        LDA (SCRPTR),Y
        DEC SCRPTR+1
        DEY
        STA (SCRPTR),Y ;TRASCRIBE PAGINA
        INY
        LDA (COLPTR),Y
        DEC COLPTR+1
        DEY
        STA (COLPTR),Y
        INC SCRPTR+1    ;INCR PUNTEROS P 0
        INC COLPTR+1    ;UNA VEZ MAS
        DEX
        BNE AGAIN
;
;++ PRODUCE BYTES EXTRA ++
        LDY #$01
ANTHER
        LDA (SCRPTR),Y
        DEY
        STA (SCRPTR),Y
        INY
        LDA (COLPTR),Y
        DEY
        STA (COLPTR),Y
        INY
        INY
        CPY #EXTRA
        BCC ANTHER      ;SI Y<EXTRA REPETIR
;
;++++ INSERTA COLUMNA DER DE PANT ++++
;
        LDA MEMLO
        STA MEMPTR
        LDA MEMHI       ;ESTABLECE PAGINA 0
        STA MEMPTR+1    ;PUNTEROS A LA MEMORIA
        LDA #NMCOLS-1
        STA SCRPTR      ;ESTABLECE PAGINA 0
        LDA #SCRNHI     ;PUNTEROS A LA PANTALLA
        STA SCRPTR+1
;       JSR COPY40      ;COPIA COLUMNA
;++++ INSERTA COLUMNA DER DE COLOR ++++
;       LDA MEMLO
        CLC
        ADC COFFLO      ;SUMA DESPLAZ AL
        STA MEMPTR      ;MAPA COLOR
        LDA MEMHI       ;ESTABLECIENDO LOS
        ADC COFFHI      ;PUNTEROS PAGINA 0
        STA MEMPTR+1
        LDA #NMCOLS-1   ;ESTABLECE PUNTEROS
        STA SCRPTR      ;PAGINA 0 AL COLOR
        LDA #COLRHI     ;RAM
        STA SCRPTR+1
        JSR COPY40      ;REALIZA LA COPIA
;
;++++ POSICIONES DEL 6 AL 0 PARA DESPLAZ ++++
;
        LDX #$06
MORE 1
        LDA SCRLRG
        AND #$F8
        STA SCRLRG
        TXA
        CLC
        ADC SCRLRG
        STA SCRLRG
;
        LDY DELAY       ;CUENTA ATRAS
MORE2
        DEY             ;VALOR RETARDO
        BNE MORE2
        DEX
        BPL MORE1
;
;++++ INCREMENTA PUNT. MEMORIA ++++
;
        LDA MEMLO
        CLC
        ADC #$01
        STA MEMLO
        LDA MEMHI
        ADC #$00
        STA MEMHI
;
;++++ COMPRUEBA FIN AREA MEMORIA ++++
;
        PLA
        TAY             ;RECUPERA LOS REG X,Y
        PLA
        TAX             ;DE LA PILA
        DEY
        BEQ NOJMP
        JMP START
NOJMP
        LDA MEMLO
        CLC
        ADC #$C0        ;SUMA 1000-40
        STA MEMLO       ;AL PUNTERO MEMORIA
        LDA MEMHI
        ADC #$03
        STA MEMHI
        DEX
        BEQ RETRN
        JMP NEXSCR
RETRN
        RTS
;
;++++ S/R DE COPIA A CADA 40 BYTES ++++
;
COPY40
        LDX #NMROWS
        LDY #$00
REPEAT
        LDA (MEMPTR),Y
        STA (SCRPTR),Y
        DEX
        BEQ FINISH      ;AGOTADAS TODAS LAS FILAS?

        LDA SCRPTR
        CLC
        ADC #NMCOLS     ;INCR EN 40 LOS PUNTS
        STA SCRPTR
        LDA SCRPTR+1
        ADC #$00
        STA SCRPTR+1

        LDA MEMPTR
        CLC
        ADC #NMCOLS
        STA MEPTR
        LDA MEMPTR+1
        ADC #$00
        STA MEMPTR+1
        JMP REPEAT
FINISH
        RTS
```



"Mindstorms" (Ideas geniales), de Seymour Papert, Harvester Press, 1980

Iniciamos esta recensión comentando dos reveladoras obras que examinan las interioridades de la informática: "Mindstorms" (Ideas geniales), de Seymour Papert, y "The soul of a new machine" (El alma de una nueva máquina), de Tracy Kidder

"Mindstorms"

"The gears of my childhood" (Los engranajes de mi infancia) se titula el prólogo de *Mindstorms*; en él Seymour Papert explica su libro, sus ideas y su persona. En su vehemente monólogo nos habla de la fascinación que, cuando niño, ejercieron en él los trenes de engranajes y cómo, a través de ellos, descubrió el placer de aprender; ahora continúa enamorado de la moderna encarnación de éstos: el ordenador personal en el cuarto de juegos.

> **"Un nuevo mundo de informática personal está a punto de aparecer... inseparable de la historia de quienes lo realizarán."**

El estilo de Papert y su propia personalidad son dominantes, pero el libro se halla imbuido de la personalidad y las ideas de Jean Piaget (1896-1980), el psicólogo-pedagogo que más influencia ha ejercido en los tiempos modernos. El LOGO, el lenguaje para ordenadores desarrollado y descrito a lo largo del libro, es en realidad el *hommage au maître* de Papert, un intento por formular una expresión concreta de las ideas de Piaget acerca del niño como "constructor activo de sus propias estructuras intelectuales".

> **"Empleo la imagen de Mathland para desarrollar mi idea de que la presencia del ordenador podría conllevar la unión de las culturas humanista y matemático-científica."**

En este sentido, el libro no versa en realidad ni sobre LOGO ni sobre Piaget, ni siquiera sobre el propio Papert; su verdadero tema es cómo los ordenadores pueden crear espacios de aprendizaje ("micromundos") en los cuales el niño pueda aprender a pensar y razonar con la misma alegría y la misma riqueza con que lo hizo Papert con sus engranajes alegóricos.

Al igual que en el caso de Piaget y su obra escrita, Papert y el LOGO han sido alabados por sus difusores y luego criticados por los revisionistas en el espacio de apenas unos pocos años. Las ideas de Piaget acerca de los distintos estadios del desarrollo del niño han sido cuestionadas por su aparente determinismo; mientras que en la actualidad hay quienes piensan que el LOGO es útil sólo para enseñar geometría y programación, en vez de considerarlo como la "piedra filosofal".

El título, *Mindstorms* (Ideas geniales), describe y define el libro. Sin duda alguna, Seymour Papert escribe tal como piensa: en una mezcla maravillosamente estimulante de frío análisis académico y ardoroso tono profesoral.

"The soul of a new machine"

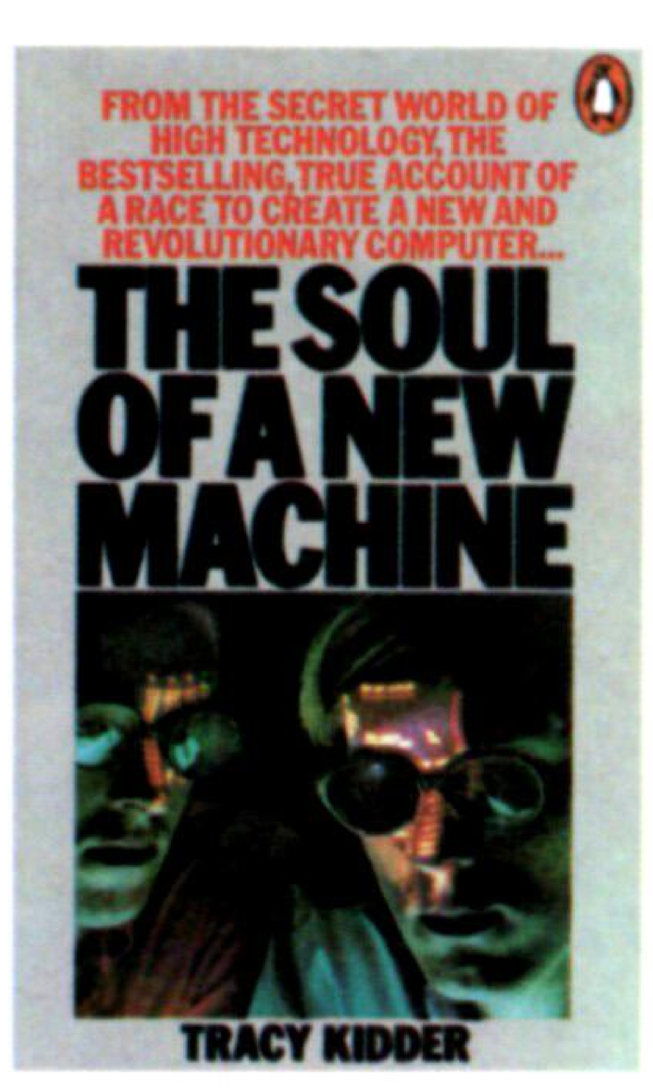
Tracy Kidder nos cuenta la historia secreta del desarrollo del miniordenador Eagle, de Data General, empresa llevada a cabo a partir de cero en el transcurso de apenas un año. El libro obtuvo el premio Pulitzer de 1982 para obras ajenas al género de ficción y ha servido de base a una película. Posee misteriosas insinuaciones psicológicas para los seguidores de Piaget, desde los rostros con cierto aire de robot de los jóvenes científicos de la cubierta del libro, pasando por los ecos edípicos de los esfuerzos de la joven empresa de ordenadores por superar a su gigantesca casa madre, DEC, hasta la figura de caracteres novelescos de West, el ingeniero de proyectos y jefe del equipo. Se la podría definir como la versión informática de *Moby Dick*, la novela de Melville, una historia que cautiva al lector, narrada con sencillez.

> **"Monótonamente, el ordenador... estaba contando una antigua historia: el materialista cuento de hadas hecho realidad."**

El libro es técnicamente espléndido, con descripciones gráficas de cada una de las etapas del diseño y la construcción del miniordenador de 32 bits. Kidder pudo observar directamente gran parte del desarrollo del proyecto y relata, en una prosa austera y cristalina, las explicaciones para las decisiones adoptadas y las acciones emprendidas.

El libro versa fundamentalmente sobre la existencia brillante y algo aislada de los técnicos cuyo microcosmos está compuesto por los sistemas operativos que ellos mismos inventan. Kidder se siente igualmente fascinado por sus reacciones personales ante el desarrollo de la máquina, por la enmarañada atmósfera que envuelve al grupo, en que se mezclan lealtad, presiones, manipulación, y por la bizantina política de la empresa, que determina la planificación y el diseño de la máquina. El pararrayos de toda esta energía es Tom West, a quien en el prólogo de la obra se describe literalmente como "un buen hombre en medio de una tormenta". Su equipo exclusivo de jóvenes ingenieros trabaja toda la noche, inspirado en parte por su ejemplo, a pesar de lo cual él cínicamente les niega una máquina de prueba esencial porque "no pagar las horas extras sale más barato que el nuevo equipo". West bien puede ser el capitán Achab o el capitán América... Tracy Kidder no parece tener muy claras las ideas en este sentido. Pero su libro arroja luz sobre algunos de estos hombres.

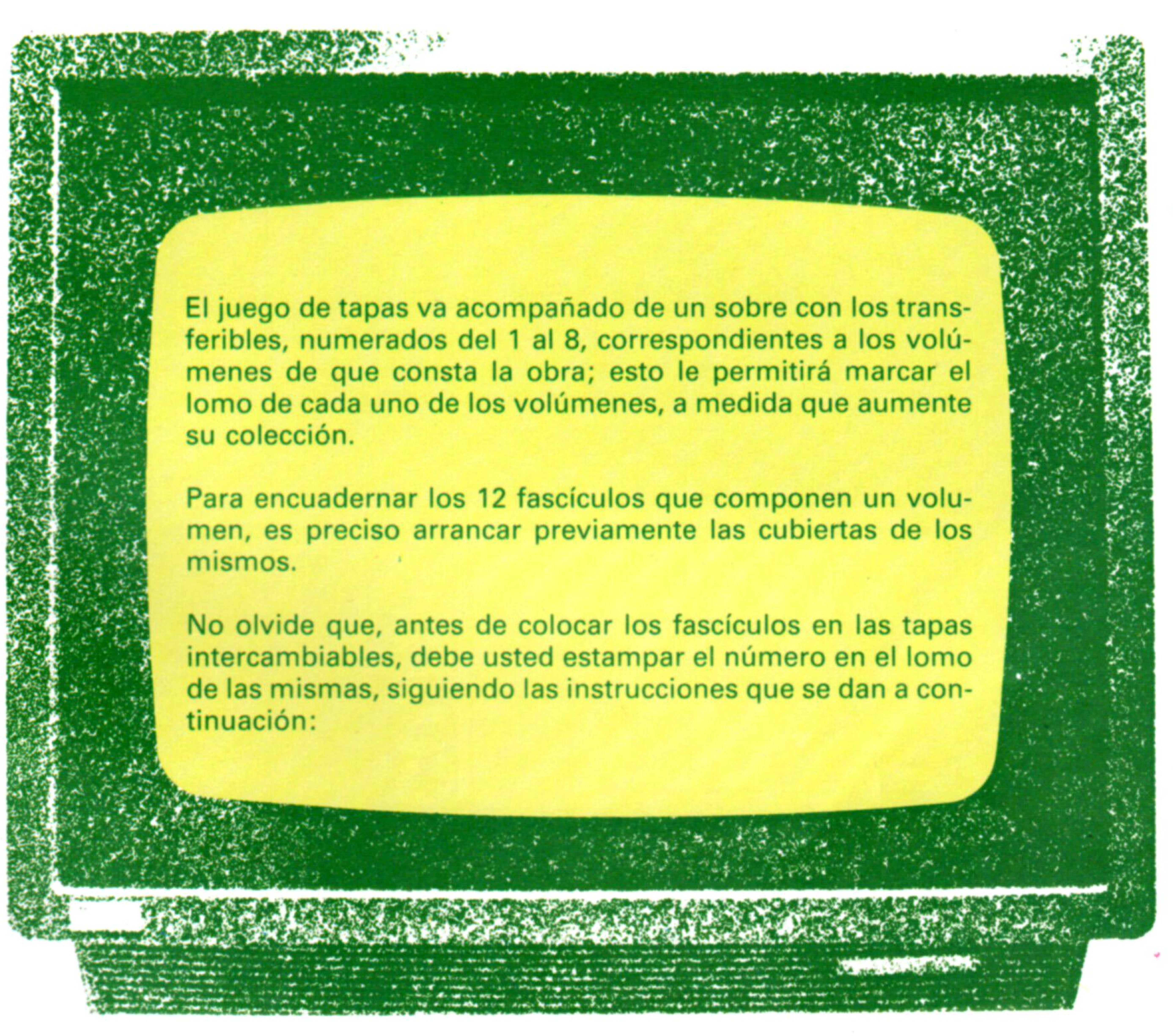
> **"Ahora el juego era diferente... La máquina ya no pertenecía a quienes la habían creado."**



"The soul of a new machine" (El alma de una nueva máquina), de Tracy Kidder, Penguin, 1982

mi COMPUTER

Con el próximo fascículo se pondrán a la venta las tapas correspondientes al sexto volumen.





1 Desprenda la hojita de protección y aplique el transferible en el lomo de la cubierta, haciendo coincidir los ángulos de referencia con los del recuadro del lomo.

2 Con un bolígrafo o un objeto de punta roma, repase varias veces el número, presionando como si quisiera borrarlo por completo.

3 Retire con cuidado y comprobará que el número ya está impreso en la cubierta. Cúbralo con la hojita de protección y repita la operación anterior con un objeto liso y redondeado, a fin de asegurar una perfecta y total adherencia.

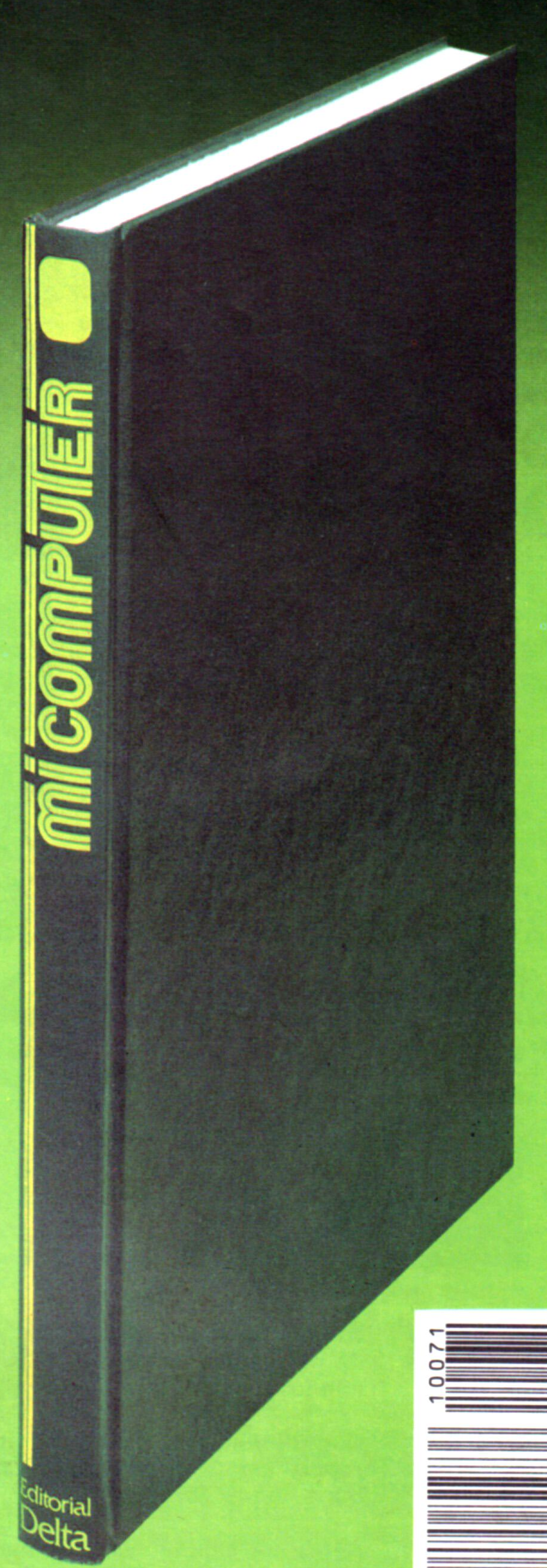
Cada sobre de transferibles contiene una serie completa de números, del 1 al 8, para fijar a los lomos de los volúmenes. Ya que en cada volumen sólo aplicará el número correspondiente, puede utilizar los restantes para hacer una prueba preliminar.

Editorial Delta

Con el próximo número se ponen a la venta las tapas intercambiables para encuadernar 12 fascículos de

mi computer

Cada juego de tapas va acompañado de una colección de transferibles, para que usted mismo pueda colocar en cada lomo el número de tomo que corresponda





10071
9 788485 822836